कर्मण्येवाधिकारस्ते मा फलेषु कदाचन ।
मा कर्मफलहेतुर्भूः मा ते संगोऽस्त्वकर्मणि ।।

You have a duty, you have an obligation, but you have no right to expect a particular result or fruit to follow from what you do.

Bhagwad Gita: Krishna explaining the working of the universe to Arjuna on the battlefield

University of Alberta

Advances in *p*-Cycle Network Design

by

Adil Abraham Kodian ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

Department of Electrical and Computer Engineering

Edmonton, Alberta

Spring 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-14002-X
Our file Notre référence
ISBN: 0-494-14002-X

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Dedicated to my parents, Dr. and Mrs. Kodian Chacko Abraham,

You raised me to believe that I can accomplish anything I decide upon.

This thesis is directly your success and your greatest gift to me.

Abstract

p-Cycles are a recent innovation in optical network protection. *p*-Cycles use pre-connected cycles of spare capacity to restore affected working traffic. In this thesis, we present four main advances to the state of the art in *p*-cycle networking. In our first project we look at the problem of migrating from an existing ring-based transport network to *p*-cycles. We show, with real-world network cases, how a carrier can migrate to *p*-cycles without affecting current customers, and gain significant increases in network carrying capacity at little or no additional capital cost. In the second study, we propose methods to incorporate precise path length restrictions in *p*-cycle design. The issue was that previously such limits could only be implemented in mesh network design. With these tools we were able to demonstrate the existence of a threshold hop limit phenomenon in *p*-cycle network design. A useful but somewhat counter intuitive result in this second study is that, in *p*-cycles, limiting the cycle size is an effective surrogate for the more complex technique of precise path length limitation in design. In our third study we propose new methods and strategies to support multiple qualities of service classes in a static *p*-cycle based network design, using the same global set of resources as required to operate a network with only a single failure protected service class. In this study we show how a *p*-cycle network operator can support discriminated service classes; such as a pre-emptible class and dual failure protected class, without the need for real-time reconfiguration in the network. Our final and most significant contribution is the design and development of Failure Independent Path Protecting *p*-Cycles. Failure Independent Path Protecting *p*-cycles are an advantageous alternative for the survivability mechanism used in current path protected optical networks. At no additional cost, they provide features such as failure independence and full pre-cross-connection.

Acknowledgements

It is an honor to call myself a Ph.D. graduate from TRLabs. I owe a debt of gratitude to a lot of people who have shaped my career and have helped me get this far.

First, I thank my wife, **Meghna**, whose love and support have helped me every day. Thanks also for being there whenever I needed a sounding board and an anchor into the real world. Thanks to your immense patience and ability to listen (which I, so far, have not learned), you probably know as much about survivable networking as anyone here in my research group.

Thanks also to **Dr. Wayne Grover**, my supervisor and mentor at TRLabs. Wayne, you have instilled in me your dedication to hard work, demanding absolute perfection from myself and from everyone who works for me and the ability to see through mountains of information and grasp true knowledge. I owe a large part of my success, both in my research and in my career to your teaching. I wish you the best and hope that other students may learn as much as I did from you, for many years to come.

Next in line to thank is **Dr. John Doucette**. Thanks John for your patience and great advice and countless hours spent on your whiteboard discussing fantastic and often unrealistic ideas. It was indeed a pleasure working with you on the DRCN 2005 paper, and I hope to work with you in the future.

I am also grateful to **TRLabs** and its past and present staff, especially **Linda Richens, Luke Chong, Rhoda Hayes, Faith Goller, Dr. Vinod Ratti, Tara Heron**, and **Chris Fields**, for providing me support and a highly professional work environment. I will carry the TRLabs name on with pride in all spheres of my activities. Thanks Luke and Vinod for starting my career off as an IT consultant to TRLabs and TRNet. Linda, you are the most efficient and hard-working administrator I have ever worked with – you kept the network systems group working by helping us cut through red-tape and paperwork many times. I also thank my current and past colleagues, especially **Anthony Sack, Dimitri Baloukov, Aden Grue, Peter Giese, Brian Forst, Dr. Dion Leung, Dr. Gangxiang Shen, Grace Shi, Govindkrishna (Govind) Kaigala, Dr. Matthieu Clouqueur** and **Ponnampalam Nakeeran**. Anthony, you are one of the smartest people I have worked with. I wish you all the success in your career and hope to work with you again someday. I especially enjoyed working with you (and Wayne) and winning the best paper award for the BROADNETS paper, which for me was technical writing 101. Dimitri, it is a pleasure to work with you on FIPP *p*-cycles. I hope to continue doing great work with you and wish you all the luck in your music and research career.

I also thank **Dr. Bruce Cockburn, Dr. Ray DeCorby, Dr. Ioanis Nikolaidis** and **Dr. Meir Herzberg** for their valuable comments and feedback as members of my Ph.D. committee.

I also want to thank **Rohit and Radhe Gupta** and all the staff at the Rohit Group of. Companies, especially **Vaneeta Patel, Krishna Gupta** and **Lesley Hudie**. Thank you for your understanding and giving me the time off from work to write this thesis. Rohit, Radhe, in these short ten months that I have been with you, I have learned more about business and ethics than any school could have taught me. This has directly and positively affected the writing of this thesis. I hope to sincerely contribute to your success for many years to come.

I also want to thank my parents in law **Col. Darab Singh** and **Mrs. Nisha Singh** for their unwavering support and encouragement.

Finally I want to thank my mom and dad and sister – **Mrs. Annie Abraham, Dr. K.C. Abraham**, and **Dr. Aldie Sanjay** without whose constant encouragement and dedication and sacrifice, I would have not reached this far. I am here mom and dad because you have raised me to believe that I can accomplish any goal I set my mind on. This is the goal.

Adil Abraham Kodian

May 15th, 2006

Table of Contents

Chapter 1. INTRODUCTION 1

1.1. Communication Network Architecture 2

1.2. Network Failures 4

1.3. What Is Survivability? 6

1.4. Motivation and Goals 8

1.5. Thesis Outline 10

Chapter 2. TRANSPORT NETWORKING FUNDAMENTALS 13

2.1. Technology at the Network-level – Layering 13

2.2. Technologies at the 'Node' - Switching 15

2.2.1. Add Drop Multiplexers (ADM) 15

2.2.2. Digital and Optical Cross Connects 16

2.3. Technology at the Channel – SONET, GigE 18

2.3.1. SONET/SDH 18

2.3.2. Generic Framing Procedure (GFP), Virtual Concatenation (VCAT), Link Capacity Adjustment Scheme (LCAS) 21

2.3.3. Gigabit Ethernet (GigE), 10 GigE 22

2.4. Wavelength Division Multiplexing 22

2.4.1. Coarse and Dense WDM 23

2.4.2. OEO Versus OOO 23

2.4.3. Routing and Wavelength Assignment (RWA) 24

2.5. Optical Service Channel, Digital Wrapper 25

2.5.1. Optical Service Channel (OSC) 25

2.5.2. Digital Wrapper 25

2.6. Fundamentals of Network Survivability 26

2.7. Physical and Logical Survivability 28

2.8. Physical Layer Techniques 29

2.8.1. Automatic Protection Switching (APS) 29

2.8.2. Self-Healing Rings 30

2.8.3. Recent Advances 34

2.9. Logical Layer Techniques 35

2.9.1. Span-Restorable Mesh Networks 35

2.9.2. Path Restorable Mesh Networks 37

2.10. *p*-Cycles 38
  2.10.1. Flow *p*-Cycles (Adapted From [GrShUSPT] and [ShGrICC03]) 41
2.11. Shared Backup Path Protection (SBPP) 46
2.12. Pre-Cross-Connected Trails (PXTs) 49
2.13. Demand-Wise Shared Protection 51
2.14. Shared Risk Link Groups (SRLG) and Dual Failures 53
2.15. Protection vs. Restoration vs. Pre-Planning vs. Pre-Connection 53
2.16. Summary 54

Chapter 3. GENERAL BACKGROUND REVIEW: SETS, GRAPHS, OPTIMIZATION 56
3.1. Set Theory and Related Concepts 56
  3.1.1. Set Operations 57
3.2. Graph Theory 57
3.3. Computational Complexity 62
  3.3.1. Asymptotic Notation 63
  3.3.2. *P* and *NP* 63
3.4. Route Finding Algorithms and Related Topics 64
  3.4.1. Shortest Path Dijkstra 64
  3.4.2. Bhandari's Variations on the Dijkstra Algorithm 65
  3.4.3. K-Shortest Paths - All Distinct Routes 65
  3.4.4. Cycle Finding, Selection and Depth First Search 66
3.5. Operations Research and Transport Network Design 68
  3.5.1. Linear Programming 69
  3.5.2. Integer Linear Programming 71
  3.5.3. Branch and Bound Heuristic 72
  3.5.4. MIPGAP 72
  3.5.5. Benefits and Limitations of ILP-based Methods 73
  3.5.6. Modularity and Economy of Scale 73
3.6. Summary 74

Chapter 4. OPTICAL NETWORKS AND PRE-CONNECTION 75
4.1. Optical Transmission System 76
  4.1.1. The Transmitter 76
  4.1.2. The Receiver 80
  4.1.3. Amplifiers and Regenerators 82
  4.1.4. Other Components 83

4.2. Signal Degradation in an Optical Fiber 84
  4.2.1. Attenuation 85
  4.2.2. Dispersion 85
4.3. Engineering of a Single Point-to-point Optical Link 87
  4.3.1. Power or Link Budget 89
  4.3.2. Selection of Fiber and Light Source 89
  4.3.3. Receiver Power Sensitivity 89
  4.3.4. Losses Due To Connectors and Splices 90
  4.3.5. Amplifiers 90
  4.3.6. Safety Margin 90
  4.3.7. Example Power Budget Calculation 90
  4.3.8. Dispersion Budget 93
  4.3.9. Network-level Challenges 93
4.4. Observations 94
4.5. Optical Layer Pre-Connection 95
4.6. Summary 96

Chapter 5. RING MINING TO *P*-CYCLES AS A TARGET ARCHITECTURE 97
5.1. Introduction 97
  5.1.1. Motivation and Goals 98
  5.1.2. Prior Work on Ring to Mesh Migration 98
  5.1.3. The Straddling Span Interface Unit (SSIU) 99
  5.1.4. Implementing *p*-Cycles on a SONET Ring Network using SSIUs 101
5.2. Ring-Mining Design Models 104
  5.2.1. Ring Mining Without Capacity Addition 104
  5.2.2. Ring Mining With Strategic Capacity Addition 106
  5.2.3. Ring Mining Without Re-Arranging Existing Traffic 107
  5.2.4. Ring-Mining With Strategic Capacity Addition and No Rearrangement of Existing Traffic 108
5.3. Experimental Setup 109
  5.3.1. Initial Ring Network Designs 109
5.4. Results and Discussion 111
  5.4.1. Ring Mining Without Capacity Addition 111
  5.4.2. Ring Mining With Strategic Capacity Addition 113
  5.4.3. Ring Mining With Strategic Capacity Addition Without Affecting Current Working Traffic 115

5.4.4. Ring-Mining With Strategic Capacity Addition With Possible Rearrangement of Working Traffic 115

5.5. Summary 116

Chapter 6. PATH LENGTH CONSTRAINTS IN *P*-CYCLE NETWORK DESIGN 117

6.1. Introduction 117

6.1.1. Overview of Path Length Restrictions 119

6.1.2. Circumference vs. Hop Limits For *p*-Cycles 119

6.1.3. Motivation and Current Goals 121

6.1.4. Threshold Hop Limit Effects in Mesh Networks (Herzberg-Bye) 121

6.1.5. Hamiltonian *p*-Cycles 122

6.1.6. The Potential Efficacy of Path Length Restrictions in *p*-Cycle Based Networks 123

6.2. ILP Design Models 124

6.2.1. Adding Path Length Constraints 124

6.3. Experimental Setup 129

6.3.1. Test Networks 129

6.3.2. Common Parameters 129

6.3.3. Mesh-Restorable Reference Designs 130

6.4. Results 131

6.4.1. Threshold Hop-Limit Effect 131

6.4.2. Comparing To Circumference-Limited Designs 133

6.4.3. Non-Joint, Non-Modular Design 134

6.4.4. Bi-Criteria Techniques 136

6.5. Summary 137

Chapter 7. MULTIPLE QUALITY OF PROTECTION *P*-CYCLE NETWORK DESIGN 139

7.1. Introduction 139

7.1.1. Motivation and Goals 139

7.2. Prior Literature 141

7.3. Multi QoP Stack 142

7.4. Preplanning and real-time operation of multi-QoP 144

7.5. Dual Failure Protection in *p*-Cycle Networks 146

7.5.1. Relationship to Prior Work 146

7.5.2. Straddling Span Protection Principle 148

7.5.3. Dispersal Protection Principle 150

7.6. ILP-based Design 152

7.7. Experimental Setup 156

7.7.1. Modeling The Dispersal Principle 156

7.7.2. Test Networks 157

7.7.3. Computational Details 158

7.8. Results 158

7.8.1. Joint Modular Design With Only Gold and Platinum 158

7.8.2. Experiments on Larger Networks 161

7.9. Summary 164

Chapter 8. FAILURE INDEPENDENT PATH PROTECTING *P*-CYCLE NETWORKS 165

8.1. Introduction 165

8.2. Background 167

8.2.1. Path Protection, Restoration and Mutual Capacity 167

8.2.2. Flow *p*-Cycles, a First Attempt 175

8.3. FIPP *p*-Cycles 176

8.3.1. Transport Network Survivability Technology "Wish-List" 176

8.3.2. FIPP Concept 177

8.3.3. FIPP *p*-Cycle Operation 180

8.3.4. FIPP *p*-Cycles and Non-Disjoint Routes 184

8.4. FIPP *p*-Cycle Network Design 186

8.4.1. Spare Capacity Placement Design Model (FIPP-SCP) (First Principle) 187

8.4.2. Joint Working and Spare Capacity Placement Design Model (FIPP-JCP) 190

8.4.3. Disjoint Route Set Approach To FIPP *p*-Cycle Network Design (Second Principle) 196

8.5. Experimental Setup 202

8.5.1. Experimental Setup For FIPP-SCP ILP Model 202

8.5.2. Experimental Setup For FIPP-SCP-DRS ILP Model 204

8.5.3. Experimental Setup For FIPP-JCP-DRS ILP Model 207

8.5.4. Test Networks and Computational Details 209

8.6. Results for the FIPP-SCP Model 211

8.6.1. Preliminary Comments 211

8.6.2. Discussion 212

8.6.3. Inspection of FIPP Design Characteristics 216

8.7. Results for the FIPP-DRS-SCP Model 219
    8.7.1. Improvements 220
8.8. Results for the FIPP-JCP-DRS Model 222
    8.8.1. Varying Cycles Per DRS 222
    8.8.2. Varying Maximum Number of Routes In Each DRS 224
    8.8.3. Varying The Number of Eligible Routes Per Demand 224
8.9. Summary 230

Chapter 9. CLOSING DISCUSSION 232
9.1. Summary of Thesis 232
9.2. Main Contributions 234
9.3. Publications and Presentations 235
    9.3.1. Presentations/Reports/In Preparation 237
9.4. Topics for Further Research 238
    9.4.1. Envelope-Based Operation of FIPP *p*-Cycle Networks 238
    9.4.2. Multi-QoP in FIPP *p*-Cycle Networks 239
    9.4.3. Dual Failure Analysis of FIPP *p*-Cycle Networks 239
    9.4.4. Ring Mining to FIPP *p*-Cycles 239
    9.4.5. Simplifying FIPP ILP-based Design Using Column Generation 239

FOOTNOTES ....

APPENDIX

APPENDIX 1. AMPL Models ....
APPENDIX 2. Test Networks (Topology and Demand Files) ....

List of Tables

Table 1. Digital Bitrates 19

Table 2. Standard SONET data rates 20

Table 3. Max Uniform Demand Growth Multiplier. (Adapted from [KoGrNFOEC03]) 112

Table 4. Capacity addition required with change allowed to working routing. (Adapted from [KoGrNFOEC03]) 114

Table 5. Capacity Addition required with no change allowed to working routing. (Adapted from [KoGrNFOEC03]) 116

Table 6. Absolute and Relative rates of spare capacity growth versus demand growth factor for different multi-QoP mixtures. (Adapted from [KoGrBROADNETS05]) 164

Table 7. Ideal Transport Network Wish list. (Adapted from [GrKoUSPT]) 177

Table 8. Demand matrix for test network. (only non-zero demands listed) (Adapted from [KoGrJLT]) 204

Table 9. FIPP-DRS-SCP Generate DRS heuristic algorithm. (Adapted from [KoGrDRCN05]) 205

Table 10. Modified GenerateDRS algorithm for the FIPP-DRS-JCP model (Adapted from [KoBaPNET06]) 208

Table 11. Total spare capacity results (channels required) for 7n12s network designs (Adapted from [KoGrJLT]) 213

Table 12. Results (spare capacity channel counts) for Cost 239 and 15n20s networks (Adapted from [KoGrJLT]) 214

List of Figures

Figure 1-1. Access, Metro and Long-Haul Networks. (Adapted from [Lt01]) 3

Figure 2-1. OSI Network Model (Adapted from [DoPhD04]) 14

Figure 2-2. Transport Network Layering (Adapted from [DoPhD04]) 15

Figure 2-3. Add Drop Multiplexer (Adapted from [KoGrNFOEC03]) 16

Figure 2-4. Digital/Optical Cross Connect 17

Figure 2-5. Simple 1+1 APS system. (Adapted from [DoPhD04]) 29

Figure 2-6. Unidirectional Path Switched Rings. (Adapted from [Grov03]) 31

Figure 2-7. 4-Fiber Bidirectional Line Switched Rings. (Adapted from [Grov03]) 32

Figure 2-8. Span restoration in a mesh network (a) Mesh Topology, (b) Span failure and (c) Restoration. (Adapted from [Grov03]) 35

Figure 2-9. Comparison of Span and Path restoration. (Adapted from [Grov03]) 38

Figure 2-10. *p*-Cycle showing (a) On-cycle span failure and (b) Straddling span failure. (Adapted from [Grov03]) 39

Figure 2-11. Examples of (a) a span protecting *p*-cycle and (b) the same cycle viewed as a flow-protecting *p*-cycle. (Adapted from [ShGrJSAC03]) 42

Figure 2-12. Various flow-to-cycle relationships that complicate flow *p*-cycles. (Adapted from [ShGrJSAC03]) 44

Figure 2-13. Flow-*p*-cycle operation (a) Typical set of flow-*p*-cycles, (b) Protection of segment A-O-N-F on failure of span ON, (c) Protection of segment A-O-M-X-Y-D for the failure of AO and (d) Protection of segment C-A-O-N for failure of span AO. (Adapted from [GrKoUSPT]) 46

Figure 2-14. Shared Backup Path Protection illustration showing a set of four working routes that can share spare capacity (Adapted from [KoGrJLT]) 48

Figure 2-15. (a) PXT backup routes for working routes shown in Figure 2-14(b) and (b) Corresponding Pre-cross-connected Trails. (Adapted from [GrKoUSPT]) 51

Figure 3-1. Graphical representation of the Level3 North Americal LHN. (Adapted from [Level3]) 58

Figure 3-2. (a) a bi-connected and two-connected network, (b) a two-connected network (but not bi-connected), (c) a connected graph 62

Figure 3-3. Transportation Problem Network Graph (Adapted from [CaOR02]) 71

Figure 4-1. Example of a 10 Gbps point-to-point DWDM link. (Adapted from [Maxim]) 77

Figure 4-2. Example of a 10Gbps DWDM transmitter (Adapted from [Maxim]) 80

Figure 4-3. Example of a 10Gbps DWDM receiver (Adapted from [Maxim]) 81

Figure 4-4. UNI, ONI, NNI for interoperability in optical networks. 94

Figure 5-1. *p*-Cycle Straddling Span Interface Unit. (Adapted from [ClGrDRCN01]) 100

Figure 5-2. Ring to *p*-Cycle conversion - Legacy Network. (Adapted from [Grov03]) 102

Figure 5-3. Ring to *p*-Cycle conversion - First Planned Growth Stage. (Adapted from [Grov03]) 103

Figure 5-4. Ring to *p*-Cycle conversion - Second Planned Growth Stage. (Adapted from [Grov03]) 103

Figure 5-5. Ring to *p*-Cycle conversion - Third Planned Growth Stage. (Adapted from [Grov03]) 104

Figure 5-6. Three network graphs used to produce the input ring designs. (Adapted from [Morley01]) 109

Figure 5-7. Isomorphic Graph of the Telus Calgary Metro Network (a) Actual physical facility and (b) Logical connections 111

Figure 5-8. Capacity Addition Profile for two test networks growing under ring mining. (Adapted from [KoGrNFOEC03]) 113

Figure 6-1. Existence proof example of potential benefits of explicit path length restriction versus cycle size limiting. (Adapted from [KoSaBROADNETS04]) 123

Figure 6-2. Left and right path definitions for (a) on-cycle spans and (b) straddling spans. (Adapted from [KoSaBROADNETS04]) 125

Figure 6-3. Test networks (a) 13n23s (501 cycles), (b) 15n26s1 (871 cycles). (Adapted from [KoSaBROADNETS04]) 129

Figure 6-4. Test networks (a) 12n19s (127 cycles), (b) NSFNET (139 cycles). (Adapted from [KoSaBROADNETS04]) 129

Figure 6-5. Total cost of capacity versus hop limit for hop-limited mesh and *p*-cycle network designs. (Adapted from [KoSaBROADNETS04]) 132

Figure 6-6. Hop-limited path distribution for $C=7$ and $H=6$ designs in the 13n23s network. (Adapted from [KoSaBROADNETS04]) 134

Figure 6-7. Capacity results for the 19n35s network. (Adapted from [KoSaBROADNETS04]) 135

Figure 6-8. Hop-limited path distribution for *C=7* and *H=6* designs in the 19n35s network. (Adapted from [KoSaBROADNETS04]) 135

Figure 6-9. Effect of bi-criteria objective in improving path-length-limited results for 19n35s design (*C*=7, *H*=6). (Adapted from [KoSaOSN05])............. 137

Figure 7-1. Multi-QoP class set. (Adapted from [KoGrBROADNETS05]) 143

Figure 7-2. Straddling span routing principle for dual-failure protection (a) *p*-cycle *X*, (b) first span failure, (c) second span failure. (Adapted from [KoGrBROADNETS05]).. 150

Figure 7-3. The dispersal principle for dual-failure survivability (a) *p*-cycles X and Y, (b) first failure and (c) second failure. (Adapted from [KoGrBROADNETS05]) 151

Figure 7-4. (a) 7n11s (b) 9n16s test networks and (c) 25n50s test network. (Adapted from [KoGrBROADNETS05]).. 158

Figure 7-5. Results for the Joint Modular min cost design (a) relative percentage increase and (b) total spare capacity. (Adapted from [KoGrBROADNETS05]) 160

Figure 7-6. Spare capacity requirements for non-joint, non-modular design. (Adapted from [KoGrBROADNETS05]).. 161

Figure 7-7. Results for 25n50s test case with static QoP mix and Lambda. (Adapted from [KoGrBROADNETS05]).. 163

Figure 8-1. Example of a FIPP *p*-cycle protecting a set of seven mutually disjoint primaries. The whole pre-connected protection *structure* is shared intact. (Adapted from [KoGrJLT]) 178

Figure 8-2. Different path to *p*-cycle topological relations for description of operation of FIPP *p*-cycles. (Adapted from [KoGrJLT])....................... 182

Figure 8-3. FIPP *p*-cycle protecting non-disjoint working paths.................................. 185

Figure 8-4. Cost 239 test network indicating span numbers. (Adapted from [BaCOST239]).. 204

Figure 8-5. 15n30s Test Network (used for FIPP-JCP-DRS trials) (Adapted from [DoPhD04]).. 207

Figure 8-6. (a) FIPP *p*-Cycle A in Solution for Cost 239 network with 19 Demands (SCP), (b) logical view, (c) example of potential additional loading capability. (Adapted from [KoGrJLT])..................... 216

Figure 8-7. (a) *p*-Cycle (B) FIPP *p*-Cycle Solution for Cost 239 network with 19 Demands (SCP) and (b) logical view. (Adapted from [KoGrJLT]) 217

Figure 8-8. (a) FIPP *p*-Cycle C and (b) logical view. (Adapted from [KoGrJLT])....... 218

Figure 8-9. (a) FIPP *p*-Cycle (D) and (b) logical view. (Adapted from [KoGrJLT])..... 218

Figure 8-10. (a) FIPP *p*-Cycle (E) and (b) logical view. (Adapted from [KoGrJLT]) 218

Figure 8-11. Conventional Span *p*-cycle solution for COST239 with 19 demands and 1000 eligible cycles. (Adapted from [KoGrJLT]) 219

Figure 8-12 – Total capacity costs of FIPP *p*-cycle network designs using the FIPP-CDS ILP model. (Adapted from [KoGrDRNC05]) 220

Figure 8-13 – Total capacity costs of FIPP *p*-cycle network designs with limited numbers of demand relations per CDS. (Adapted from [KoGrDRCN05]) 222

Figure 8-14. Cost Versus Cycles per DRS. (Adapted from [KoBaPNET06]) 223

Figure 8-15. Cost versus Routes Per DRS. (Adapted from [KoBaPNET06]) 224

Figure 8-16. Cost versus number of eligible routes per demand. (Adapted from [KoBaPNET06]) 225

Figure 8-17. Average Length versus Nth shortest route. (Adapted from [KoBaPNET06]) 226

Figure 8-18. Number of Eligible Routes per Demand vs. Cost, with history propagation. (Adapted from [KoBaPNET06]) 229

Figure 8-19. Complete FIPP DRS Solution with Shortest Routes Only. (Adapted from [KoBaPNET06]) 229

Figure 8-20. Complete FIPP solution when up to two shortest routes are allowed. (Adapted from [KoBaPNET06]) 230

List of Abbreviations

3R	Retiming, Regeneration, Retransmission
10GigE	10 Gigabit Ethernet
ADM	Add Drop Multiplexer
AIS	Alarm Inhibit Signal
APD	Avalanche Photo Diode
APS	Automatic Protection Switching
APWCE	Adaptive Protected Working Capacity Envelope
ATM	Asynchronous Transport Mode
BDP	Bandwidth Distance Product
BER	Bit Error Rate
BFS	Breadth First Search
BLSR	Bidirectional Line Switched Rings
CDS	Compatible Demand Set
CL	Circumference Limited
CO	Central office
CRTC	Canadian Radio-television and Telecommunications Commission
DCF	Dispersion Compensated Fiber

DCS	Digital Crossconnect Switch
DEMUX	Demultiplexer
DFS	Depth First Search
DPP	Distributed Pre-Planning
DRS	Disjoint Route Set
DS	Digital Signal (such as DS-1, DS-3)
DSF	Dispersion Shifted Fiber
DSL	Digital Subscriber Line (also Asynchronous DSL)
DSP	Demand-wise Shared Protection
DW	Digital Wrapper
DWDM	Dense Wavelength Division Multiplexing
EAM	Electro-Absorption-Modulator
EDFA	Erbium Doped Fiber Amplifier
FCC	Federal Communications Commission
FCRIP	Fixed Charge Routing IP
FEC	Forward Error Correction
FIPP	Failure-Independent Path Protecting *p*-Cycles
FOM	Figure of Merit
FTTC	Fiber-To-The-Curb

FTTH	Fiber-To-The-Home
GFP	Generic Framing Procedure
GigE	Gigabit Ethernet
GMPLS	Generalized Multi Protocol Label Switching
HL	Hop Limited
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
ISP	Internet Service Provider
JCA	Joint Capacity Assignment
JCP	Joint Capacity Placement
LAN	Local Area Network
LCAS	Link Capacity Adjustment Scheme
LED	Light Emitting Diode
LHN	Long Haul Network
LoS	Loss of Signal
LP	Linear Programming
MAN	Metropolitan Area Network
MCMF	Multi Commodity Max Flow
MILP/MIP	Mixed ILP

MPLS	Multi-Protocol Label Switching
MSPR	Multiplex Switched Path Ring
MUX	Multiplexer
NEPC	Node-Encircling *p*-Cycles
NGS	Next Generation SONET
NIC	Network Interface Card
NNI	Network to Network Interface
NP	Nondeterministic Polynomial
OADM	Optical ADM
OC	Optical Channel
OD	Origin Destination (Pair)
OEO/O-E-O	Optical-Electronic-Optical (Switching)
OOO/O-O-O	Optical-Optical-Optical (Switching)
OPPR	Optical Path Protected Ring
OR	Operations Research
OSC	Optical Service Channel
OSI	Open Systems Interconnection
OSPR	Optical Shared Protection Ring
OTDR	Optical Time Domain Reflectometry

OXC	Optical Cross Connect
PDA	Personal Digital Assistant
POP	Point of Presence
POTS	Plain Old Telephone System
PPMN	Path-Protected Mesh Networks
PR	Path Restoration
PWCE	Protected Working Capacity Envelope
PWP	Pure WP
PXT	Pre Crossconnected Trail
QoP	Quality of Protection
QoS	Quality of Service
RAM	Random Access Memory
ROADM	Reconfigurable OADM
RPR	Resilient Packet Switched Ring
RWA	Routing and Wavelength Assignment
RX	Receiver
SBPP	Shared Backup Path Protection
SCIP	Span Coverage IP
SCP	Spare Capacity Placement

SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
SONET	Synchronous Optical Network
SR	Span Restoration
SRLG	Shared Risk Link Group
SRP	Spatial Re-use Protocol
SSIU	Straddling Span Interface Unit
STM	Synchronous Transport Module
STS	Synchronous Transport Signal
TCP	Transmission Control Protocol
TDM	Time Domain Multiplexing
TEC	Thermo Electric Cooling
TS	Topological Score
TU	Total Unimodularity
TX	Transmitter
UNI	User to Network Interface
UPSR	Unidirectional Path-Switched Ring
VCAT	Virtual Concatenation

VCO	Voltage Controlled Oscillator
VLAN	Virtual LAN
VOIP/VoIP	Voice Over Internet Protocol
VWP	Virtual WP
WAN	Wide Area Network
WC	Wavelength Converter
WDM	Wavelength Division Multiplexing
WP	Wavelength Path

Chapter 1. INTRODUCTION

> *"Failure is all around us. Failure is pervasive. Failure is everywhere, across time, across place and across different aspects of life."*
>
> *Paul Ormerod [Faber01],*

All man-made things fail. For a system designer, the only certainty is that the system he/she designs, will fail at some unspecified time in the future. The Internet is one of the most complex systems that man has ever invented. With over a billion users, it runs over a backbone transport network system that services not only the Internet but also cell phones, 911 calls, bank machines, leased lines, etc. Each service has a distinct virtual network built on top of a common transport networking infrastructure. The number of users of the transport network is therefore much higher than the number of Internet users. The transport network, like any other man-made system, is susceptible to failures. Failure in the transport network manifests as a simultaneous failure in apparently unrelated virtual networks that use the network and may simultaneously affect thousands or millions of users. For example, a single cable cut in the transport network may affect Internet services for some users, whereas for other users it may cut off 911 services. Lives depend on services such as 911 and air traffic control. Thus it is important that the impact of the failure is minimized. The science of Survivable Networking is based on two fundamental premises:

a) Systems can be designed with the conscious intent of reducing chances of failure; and,

b) Systems can be designed to minimize the impact of failure, when it happens.

To ensure the former, network designers take care that the best available components and industry practices are used. For the latter, designers rely on building in survivability mechanisms and equipment or system redundancy.

This thesis is comprised of four advanced studies of a transport network survivability mechanism called *p*-cycles. The ultimate aim herein is the ability to design economically viable communication backbones that survive failures so elegantly, simply and quickly that the end-user is unaware that a failure has occurred at all.

In this Chapter, we will review the architecture of the transport network and then go on to explain some of the fundamental concepts and terms in this field.

1.1. Communication Network Architecture

The broadband network that enables the Internet can be categorized into a three-level hierarchy based on size and function. Figure 1-1(adapted from [Lt01]) illustrates the architecture of the Internet. Most residential customers, whether on dial-up, Digital Subscriber Line (DSL) or on cable modems, are grouped in the access network. Regular Plain Old Telephone System (POTS) lines, corporate WANs, cellular phones, etc. are also part of the access network. Access networks are typically characterized by a wide variety of protocols and access mechanisms and usually represent the outer edge of the communication network infrastructure. All of the traffic in the access network is aggregated at a local switching office and is routed onto a larger Metropolitan Area Network (MAN). (This aggregation is a non-trivial undertaking and is a separate area of research called Traffic Grooming[i].)

MANs typically use fiber optic cables as the physical transport technology. Data rates on metro area networks can vary from the smallest being a DS1 at 1.5 Mbit/s to the high-speed OC-192 (STS-192) that supports 10 Gbit/s. MANs are the middle level in Figure 1-1. There are typically many MANs in an average sized city. Each MAN operator may have points of presence (POPs) in other competing MANs. Data that originates in one MAN and is destined for another MAN is exchanged at these POPs. Traffic that is not destined for other MANs in the same city is then aggregated onto a Long-Haul Network (LHN).

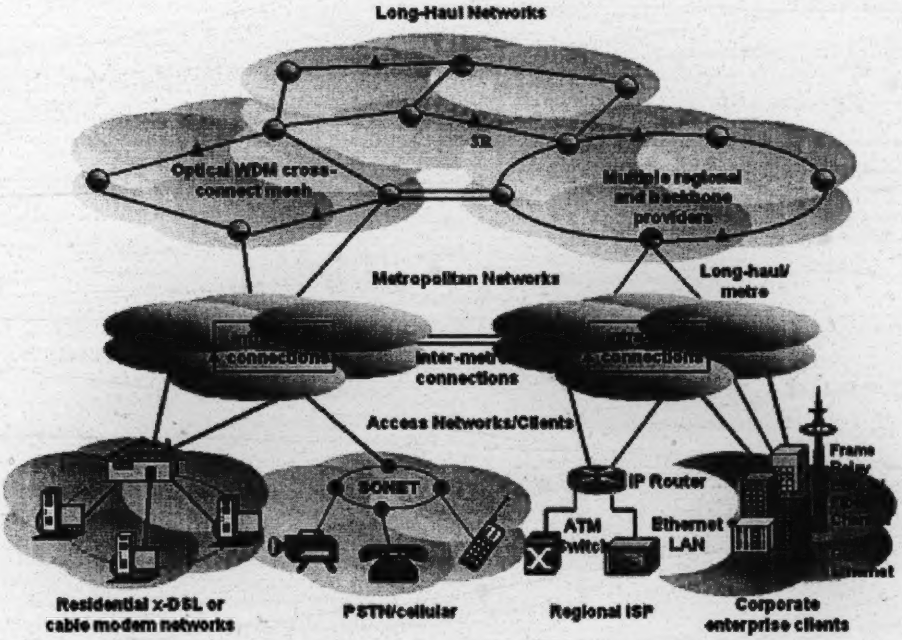

Long-Haul Networks
SR
Optical WDM cross-connect mesh
Multiple regional and backbone providers
Metropolitan Networks
Long-haul/ metro
connections
inter-metro connections
connections
Access Networks/Clients
SONET
IP Router
ATM Switch
Ethernet LAN
Frame Relay
Ethernet
Residential x-DSL or cable modem networks
PSTN/cellular
Regional ISP
Corporate enterprise clients

Figure 1-1. Access, Metro and Long-Haul Networks. (Adapted from [Lt01])

LHNs are almost completely based on fiber optic systems and usually provide a lesser variety of fixed data pipes as compared to MANs or access networks. LHNs may span thousands of kilometers and extend across continental-sized geographic areas. LHNs also extend across the Pacific and Atlantic oceans and carry intercontinental traffic on undersea cables and satellite links. Because of the cost and size of LHNs there tend to be many fewer LHN operators than MAN or Access network operators. Many countries have nationalized their LHN infrastructure because of the huge capital and operational expense involved. In the access network, the operators are concerned with the individual traffic types. In contrast, in MANs and LHNs, the main goal is to transport the data bits from one point to another without actually considering details about the services that generated them.

In this thesis, we primarily deal with problems that address issues in MAN and LHN design. We therefore are concerned with reliably transporting a number of unit granularity bit frames from one point to the other using, in some cases, the least amount of spare resources. MANs and LHNs are referred to as Transport Networks. The only difference between a MAN and a LHN is in the size and complexity. MANs are typically very dense but usually cover a small geographic area such as a city. LHNs are sparse and distributed across the globe.

1.2. Network Failures

End-users often do not appreciate how important modern transport networks really are to them. As a society we are used to a dial tone whenever we pick up the phone, 24 hour access to the 911 service, and bank machines, etc. These services essentially form the face of the transport network that is visible to the common man. At the transport networking level, however, cell-phones, computers, portable digital assistants are simply bit generating devices. The bits generated are groomed together such that bits with common destinations are packaged together in bulk. The transport network then reliably and quickly ships these bits from their point of origin to their destinations. The transport network isn't really all that different from a warehousing and supply system, except in scale and time-sensitivity. Transporting bits reliably is therefore the main purpose of a transport network.

In some transport networks that are based on microwave towers or satellite transmission systems, the network is as reliable as the individual components i.e. the reliability of satellite ground stations and microwave towers. Optical fiber technologies have largely overtaken microwave transport networks because they have incredible data carrying capacities. (A single fiber can theoretically carry an estimated 30 Tbits of information in one second). Pioneer Consulting LLC of Massachusetts, U.S.A. estimates that the total peak hour transport bandwidth requirement in all of North America is about 17.92 Tbit/s – which is less than the theoretical information capacity of a single fiber.

In a microwave or satellite communication system, it is fairly difficult to "cut" electromagnetic waves (except in the limited case of weather disturbances and magnetic storms), and redundant microwave transmission equipment that is securely protected inside an operator's premises will rarely break down. Optical fibers on the other hand are housed in cables that are routed across thousands of miles of land, over poles, underground, under-water, etc. As experience has shown, cable cuts are a fairly common and frequent occurrence. Optical network transmission and receiving equipment is also far more complex than microwave or satellite equipment and is therefore relatively less reliable. A recent study [NeHaOFC99] estimates that one mile of fiber cable will operate reliably for about 228 years before failing. But a typical fiber based LHN has hundreds of thousands of installed fiber miles. A typical fiber MAN for a city the size of Edmonton may have up to ten thousand fiber-route miles of installed fiber. The numbers work out to about one failure every day for a typical LHN and one failure every four days for a typical MAN.

Fiber cuts cause outages in many higher layer services simultaneously and therefore affect a large number of users at once and usually are front page news. A severed Sprint cable cut resulted in an outage for thousands of North Carolina customers for over five hours [FCC2],[FCC3]. In another incident, a fire melted a fiber cable affecting five thousand customers for over nine hours [FCC4],[FCC1]. 911 services to a part of San Diego were out for over four hours recently [FCC4].

Canada is also not immune to these network outages and there have been several similar high profile failures reported in the news. A maintenance worker dropped a tool into an electrical panel at Bell Canada. The resulting fire disrupted communications at the Toronto Stock Exchange, private line communications between the Canadian Embassy in Washington DC and Ottawa and thousands of bank machines, and caused an outage on the network backbones of residential Internet providers in Toronto [WIRED01]. In January 2003, a cargo ship dropped anchor and severed an underwater cable that could not be repaired for two weeks. The resulting network outage lasted four days. In 2001 all Internet traffic between Asia and North America was also cut off by an undersea cable outage [Tham01]. An MCI cable failure disrupted POTS services to much of the East Coast [BoBr98]. An AMTRAK train crash caused a fire inside a tunnel that affected the networks of most of the ISPs in the US [Mc01]. Many other such events have been reported in the news recently [We01], [QSN03]

To make things worse not all network failures are accidental. Natural disasters and deliberate human sabotage may also play their part. Earthquakes, tsunamis and hurricanes also have a profound and unpredictable effect on the communication network. For example, the recent hurricane 'Ivan' cut off Jamaica from the rest of the world as it passed through the Caribbean, by destroying undersea optical fiber cable landing stations on the beach. Some services were not restored for as long as a week. This is potentially disastrous for a country like Jamaica, where the local economy is based on communication-intensive industries like call centers, air travel and tourism. A recent article [KeWIRED] points out that the ultimate terrorist threat could be a simple backhoe. A backhoe used at just the right location could disrupt communication services to a large section of the population at once, significantly affecting the population. The economic effects of network failure are also staggering. A recent estimate by Infonetics Research is that large enterprises experience 1.76 outages per month. This translates into hourly losses that may range from 90 thousand to 6 million dollars. The Gartner Group reports that (quoting) *"Through 2004, large U.S. enterprises will have lost more than $500 million in potential revenue due to network failures that affect critical business functions."* [OPNET]. Network failures also have a varying level of impact depending on the level at which it occurs and the time at which it occurs [McDonald94]. A cut in a single T1 line, i.e. a failure in the access network will affect only a few users. In contrast a failure in the LHN in Winnipeg may take out long distance services for the west coast of Canada. Most of the research in building resilient networks is therefore directed toward enabling MANs and LHNs to survive failures. In the next section we develop the two basic concepts of Availability and Reliability.

1.3. What Is Survivability?

Before defining survivability, let us first define two related terms – **Reliability** and **Availability** that are often confused with each other. Reliability is the probability that a system stays in an operative condition, given that the system was initially operative, continuously throughout the duration of its mission, or more generally operates to time *t* without failure. Availability is the probability that a system is found operative at an arbitrary given time. Depending on the system and the application Availability and Reliability take on varying levels of importance. For example, it is a usual expectation that an aircraft is 100% reliable during its 8 hour transatlantic flight. But there is no expectation for the aircraft to fly 24 hours of the day. In the airline industry therefore a lot of attention is paid to using the most reliable engines and accessories when building the aircraft. Each aircraft undergoes regular maintenance to ensure that all individual components are in working order. In fact system reliability is so important that all critical flight components are designed to be at least 100% redundant. A transport network on the other hand has to be functional 24 hours of the day, 7 days of the week, 365 days a year, etc. A key difference between an aircraft and a transport network is that while there cannot be usually any in-flight repair of a failure in an airplane, optical equipment failures and cable cuts can be repaired. Thus any metric that measures the quality and performance of an optical network must take into account this possibility of in-service repair.

Availability is therefore the more defining metric for communication networks. What is important is that the network be found operative or available whenever it is required for use. If at three a.m. in the morning the network suffers a 200 millisecond outage, (the outage is so small because the affected traffic is immediately re-routed along alternate routes) it usually affects only a small number of users and may decrease the reliability metric while overall availability remains largely unchanged. From a mathematical perspective, reliability can only decrease with time, whereas availability stays fairly constant, provided the network has some survivability mechanism built in and has sufficient redundancy to provide for alternate routes for all affected traffic. Typically network operators provide service level guarantees that may specify, for example, that the availability of service is 99.999%. This is sometimes referred to as 'five-nines availability' (See work by Clouqueur in [Cl04] for more about network availability). This translates into an expected outage of 5 minutes per year – on average. In Canada the CRTC regulates the level of compensation that operator is required to give to clients if service outage occurs that is longer than the maximum specified in the service level agreement[ii]. High service availability is therefore a major business and regulatory requirement.

How can a network then attain near 100% availability? The network designer/planner has two possible options – a) build the network out of highly reliable components or b) perform in-service repair. Transmission equipment such as line-cards and switches can be physically protected (to increase reliability) quite easily, therefore card and switch failures are relatively rare. Similarly fiber cable is protected by adding metallic sheathing, use of deep concrete ducts, burial in the earth, mooring to the sea-floor, etc, but experience has shown that there is really no way to protect each and every mile of cable against essentially random events [Craw93] as apparently bizarre as sharks biting and severing buried undersea optical cables or as common as a backhoe digging up buried cable.

Therefore, instead of concentrating only on physical cable protection, i.e. option (a), a more strategic option is to develop 'repair' protocols and mechanisms such that when a fiber gets cut, the failed data connections can be re-established automatically through alternate routes over redundant capacity pre-planned into the network. While cut cables must still be repaired physically, repair of service in this context does not imply physical "truck rollouts" as the only manner of recovery. Instead, many techniques (to be reviewed later) provide for automatic failover to a backup route or re-routing (sometimes in under 50-200 milliseconds of detecting and localizing a failure) of affected data connections using as little as 30 to 40% more capacity than required for just routing working capacity. Physical cable repair can then be carried out while the network in this alternate working state and once the repair is complete, the re-routed services can revert back to their normal routes. The time taken to physically re-splice or reconnect the cable will generally not affect the end-user and the overall availability of service. After the cut is repaired, the network may revert to its previous state without affecting any users. Thus availability can be enhanced by either improving reliability or by building in *survivability*. **Survivability** can therefore be defined as the art of guaranteeing high system availability without relying only on high reliability of individual components.

1.4. Motivation and Goals

A recent development in transport network survivability is the *p*-cycle technique [GrStICC98], [GrStDRCN00]. It offers very fast protection switching with guaranteed transmission integrity of protection paths. *p*-Cycle designs also achieve high spare capacity efficiency and also support independent routing of traffic, without constraints arising from the placement of protection structures (current technology like rings (to be reviewed) force working traffic to follow a ring-like route as well). Another operational advantage is that *p*-cycle protection can be implemented on simple nodal equipment

[KoGrNFOEC03] and does not require expensive optical switches. *p*-Cycle configuration can be logically managed on a per channel basis, can be adapted to implement multi-priority protection, can adapt to changing traffic patterns, can be adapted for maximized survivability in multiple failure scenarios, supports path length or optical reach restrictions, etc. All factors considered, *p*-cycles offer intriguing and promising alternatives to conventional optical network architectures and there is considerable motivation to further explore and refine this domain of networking technology and theory.

Our first three studies look at span protecting *p*-cycle design problems and our final study is on FIPP *p*-cycles listed as follows:

a) Reconfiguring conventional rings to operate as *p*-cycles,

b) Incorporating strict path length restraints on *p*-cycles, in design, so as not to exceed optical reach limitations, and,

c) Simultaneously supporting multiple survivability levels for working traffic using the same set of resources.

d) Design of FIPP *p*-cycles.

Our first three studies significantly advance the state of the art in conventional span protecting *p*-cycles. However, as attractive as conventional *p*-cycles are, they do not offer end-to-end protection of light-paths, and hence do not, typically, reach efficiencies as high as some path-oriented survivability mechanisms that have been reported in the literature. They also do not offer a straightforward method to protect against equipment failure at the switching offices, but instead rely on special adaptations such as the concept of Node Encircling *p*-Cycles [DoGeDRCN05]. The other main contribution of this thesis research is the co-invention of Failure Independent Path Protecting (FIPP) *p*-Cycles, a new kind of *p*-cycle that offers failure independent, pre-connected protection of optical paths end-to-end. We will review what each of these terms mean and their importance in transport networking. Results show that FIPP *p*-cycles are in the same range of capacity efficiency as Shared Backup Path Protection (SBPP, to be reviewed), which is the current state of the art in path-oriented survivability. At the same time, however, FIPP *p*-cycles also provide optically pre-connected backup paths. FIPP *p*-cycles are therefore an attractive and equal-cost replacement for SBPP. We believe that FIPP *p*-cycles are the first path-oriented, optically pre-connected, highly efficient, failure independent transport network survivability mechanism in the literature. In the remainder of this thesis, to avoid confusion, regular *p*-cycles are referred to as conventional *p*-cycles or span *p*-cycles or *p*-cycles, while this new type of *p*-cycle is always referred to as a FIPP *p*-cycle in the Chapters to follow.

1.5. Thesis Outline

The thesis is broadly split into two sections. The first section, Chapters 1 to 4, walks the reader through all the relevant background material. Wherever possible, the discussion is kept close to the studies discussed in the later Chapters. The second section, Chapters 5 to 8, contain the main body of work done in this Ph.D. Each Chapter in this section is based on material previously published at various international journals and conferences. At the start of each Chapter in this section we list the publications that are summarized.

Chapter 1, the current Chapter, introduces the problem area and the general philosophy behind this work. **Chapter 2** is a brief introduction to transport networks and survivability. In Chapter 2 we further develop basic concepts and look at network layering principles. We also briefly review other survivability mechanisms from recent literature. **Chapter 3** is a review of graph theory, optimization and related algorithms as applied to survivable network design. We develop some understanding of routing algorithms and formalize some terminology used to represent network information in the subsequent Chapters. Chapter 3 also includes a quick overview of combinatorial optimization as applied to transport networks and briefly explores concepts such as jointness, modularity and economy of scale. **Chapter 4** presents an overview of optical networking theory. Chapter 4 is mainly intended to be a pre-cursor to Chapter 8 where we use concepts such as optical-pre-connection and failure independence to explain the benefits of using FIPP *p*-cycles. In this Chapter we develop some broad understanding of the state of the art in optical switching. We also look at some of the common optical fiber impairments and walk through a broad overview of how a single 10G point-to-point optical link may be designed.

In **Chapter 5** we present our first *p*-cycle design study where we develop new techniques for migration of a conventional ring based network to a modern *p*-cycle based network. This work was done in collaboration with the Technology Strategy Group at TELUS Communications. TELUS™ is Canada's second largest telecom service provider and operates one of the fastest growing transport networks today. This is published in [KoGrNFOEC03] and Chapter 10 of [Grov03]. "Ring-Mining" answers the question – "what level of traffic growth could the network sustain if the operating technology of the TELUS Calgary MAN shifted from rings to *p*-cycles?" An added objective of this study was to propose both the hardware and firmware changes at the central office (CO) and the new operational protocols that would optimally enable the re-use of existing ring equipment.

During this study on ring-mining, we found a problem in precisely comparing *p*-cycles to span restorable mesh networks. While limiting the maximum path length of backup paths was easily possible in mesh network design, we found that the corresponding limits were not possible to be enforced in *p*-cycles using current design algorithms, without indirectly restricting the *p*-cycle size. This motivates the collaborative study in **Chapter 6** (published in [KoSaBROADNETS04] and [KoSaOSN04]) on the effect of *p*-cycle path length constraints. In Chapter 6, we answer two open questions in *p*-cycle theory – a) Do *p*-cycle network designs exhibit a "threshold hop-limit" effect corresponding to that aspect of span-restorable mesh networks? and b) How well does simple limitation of cycle circumferences compare to a more involved design method of directly asserting a hop (or distance) limit on the maximum length of protection paths. The answers to these questions, and the methods developed to address them (discussed in **Chapter 6**), both enhance our ability to design *p*-cycle networks in which limitations to optically transparent reach (or other hop or distance limitations) can be directly taken into account.

In a competitive business with a diverse set of users and applications, it is generally desirable to be able to provide a range of service offerings in some efficient way. Ideally all differentiated services should be provided using only one common set of resources, configured as needed to match the actual service mix. In **Chapter 7** (based on [KoGrBROADNETS05] and [KoGrOSN05]) we develop theory and techniques to provide differentiated levels of survivability assurance (and hence differentiated availability guarantees) to various services through a *p*-cycle based transport network. All services are provided for within a single integrated framework for operating and planning such a multiple quality of protection network. Another question of interest also answered in **Chapter 7** is: would it be possible to provide guaranteed *dual-failure* survivability to the high-revenue or critical services (such as 911 or air traffic control) with little or no more resources than the uniformly single-failure protected network requires?

Recent innovations in photonic technology are now bringing closer to reality all-optical switching of end-to-end light-paths. In an all-optical network, the working path is routed from origin to destination completely in the light-path domain. At no stage is the data converted to the electrical domain from the optical domain (O-E-O) for regeneration or noise filtering or switching. This is attractive because current electrical processing is limited to about 40Gbit/s per wavelength, whereas in theory a single fiber may carry up to 30Tbit/s. This severe bottleneck is overcome in all-optical or transparent processing where the payloads themselves are not handled at the intermediate nodes. But engineering an optical wavelength end-to-end is a complex process that involves careful tweaking of parameters such as power levels, noise levels and amplifier gains. Dynamically creating end-to-end optical protection paths remains a difficult task even with state-of-the-art equipment. It is therefore important that the optical protection paths be in a known-working condition, prior to their use. In **Chapter 8** we present Failure Independent Path Protecting (FIPP) *p*-cycles. FIPP *p*-cycles are a fast, efficient, failure-independent, pre-connected, path-oriented, automatic and localized optical network survivability mechanism. FIPP *p*-cycles are a replacement for SBPP which is the IETF standardized state of the art path survivability mechanism. FIPP *p*-cycles are a significant advance in the state of the art in survivable network design and is the most important contribution of this thesis. **Chapter 9** is a brief summary of the thesis with a list of future projects (some of which are currently being worked on by other members in our research group.)

Chapter 2. TRANSPORT NETWORKING FUNDAMENTALS

As discussed in Chapter 1, a transport network provides the bulk data carriage for a variety of upper layer services or applications. The transport network itself is essentially agnostic about the types of data it carries. In this Chapter we present a quick overview of various background topics in transport networking and briefly introduce the reader to the technology and terminology in this field.

2.1. Technology at the Network-level – Layering

The OSI model as shown in Figure 2-1 provides a layered standard framework for the design of Internet communication systems that are compatible across different computer systems and networks [FoMcGraw98]. The OSI model consists of seven layers. Each layer is simply a virtual grouping of network functions that can define the services available to the layer above it. Layering simplifies the design of network communication software and devices. For example, when writing an application, such as a browser, the programmer who works primarily in the application layer, does not have to worry himself about technical details such as the electrical voltage levels at the output of communication devices such as an Ethernet network interface card (NIC). The interface between the various layers is well defined and the level of abstraction increases from bottom to top. This architectural framework allows designers to modify or improve functions in one layer without affecting overall communication.

Typically the lower four layers, Physical, Data-Link, Network and Transport, collectively form the logical transport network. But this is not the transport network in the current context. A common misconception is that when you send an e-mail, each resultant Ethernet data packet makes its way to its destination individually. The typical LAN's native transport protocol, Ethernet, is not suitable for direct application to a laser for transmission across long distances. The packets generated in the LAN are actually passed down to the transport network system that itself has a separate set of functional layers as shown in Figure 2-2. In one model (at the extreme left of Figure 2-2) IP packets

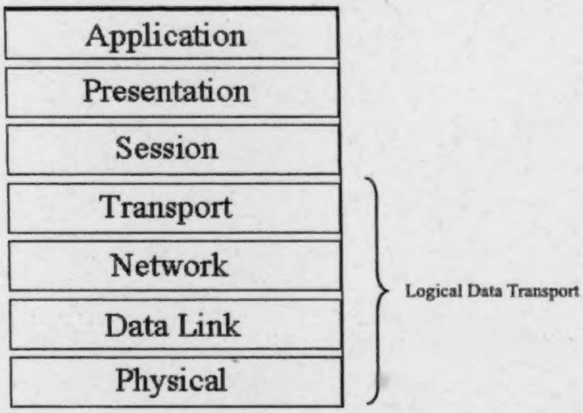


Figure 2-1. OSI Network Model (Adapted from [DoPhD04])

are transported over pre-defined end-to-end Generalized Multi Protocol Label Switching (GMPLS) data tunnels [DaReKauffman02] which use capacity on SONET [An95][An95a] Bidirectional Line Switched Rings (BLSRs) [Bell95][Bell95a] that are in turn served using Wavelength Division Multiplexing (WDM) [RaMuINFOCOM99] capable nodal equipment on physical fiber media. Each layer has a set of important functions, and similar to the structure of the OSI layer, the interface between the different layers is well defined and for the most part, standardized. The SONET layer, for example, may be responsible for survivability in the event of a fiber cut, while the GMPLS layer may have the responsibility of ensuring the end-to-end routing of the data. In general, layering decreases overall system complexity when designing transport networks by precisely defining the inter-layer communication interface. A transport network capacity planner then simply calculates the number of SONET frames required to transport the data without worrying about higher layer issues like packet queuing, delay, etc., or lower layer issues like output voltage on the card and laser center frequency stabilization.

Thus, when we talk about survivability in this thesis, we are not talking about survivability of individual phone calls or e-mail data packets, but instead are concerned with the entire bulk quantity of data generated by these individual services, surviving a failure. At this level, we are also not looking for sub-millisecond survival. As long as the optical transport network survives a failure in 100 to 200ms, the failure is transparent to the upper data network layers.

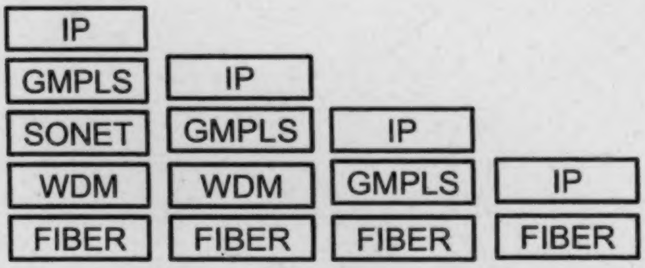


Figure 2-2. Transport Network Layering (Adapted from [DoPhD04])

2.2. Technologies at the 'Node' - Switching

In a graph theoretic representation of a transport network, a node is an abstraction for the collection of equipment that is used to interface with the network. A node includes the CO building, electrical systems, and all the switching and line termination equipment located at the CO. There are two basic types of switching elements: a) the Add Drop Multiplexer (ADM) and b) the Digital/Optical Cross Connect Switch (DCS/DXC/OXC). In this section we attempt to briefly walk through some of the key features of both these technologies. For detailed technical information and standards we rely on references that are provided. The main aim here is to develop a broad understanding of the various technologies at a transport network node.

2.2.1. Add Drop Multiplexers (ADM)

An ADM is a terminating device with only two main line rate interfaces. These are typically referred to as East and West lines. An ADM may also have local ports (sometimes referred to as the North interface) that permit it to 'drop' lower tributary rate traffic with destinations local to the ADM or to 'add' locally sourced tributaries into the outgoing interface. A typical ADM's logical diagram is shown in Figure 2-3

In 4-fiber ADM equipment, like the example shown in Figure 2-3 the East and West interfaces have two fibers each. One of the fibers is used for routing working traffic and the other is kept spare to support backup traffic, in the event of a failure. In 2-fiber ADM equipment, half the capacity of each fiber is kept reserved for protection traffic. Basically, ADMS are capable of switching data between time-slots on the four fibers connected to its interfaces. Recently wavelength converters have been added to ADM equipment to support wavelength multiplexing among tributaries. Such equipment is called an Optical ADM (OADM). Some OADMs also support dynamic run-time reconfiguration of tributary switching at the light-path layer and are called ROADMs. In this thesis, unless otherwise mentioned, we will use the term ADM to refer to both OADMs and ROADMs. Typically, from the point of view of network design, the precise details about the ADM don't really matter, as long as the assumptions made during design are feasible in some way using the available equipment.

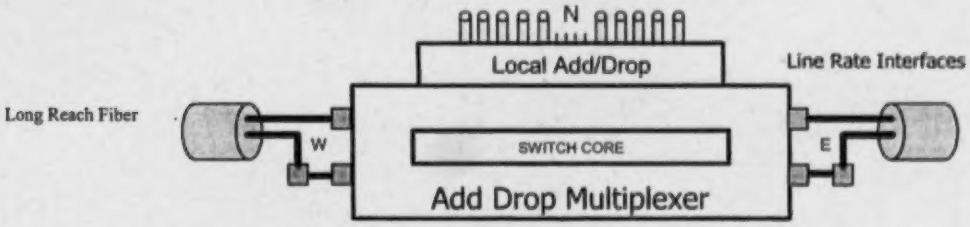


Figure 2-3. Add Drop Multiplexer (Adapted from [KoGrNFOEC03])

2.2.2. Digital and Optical Cross Connects

In the simplest of terms, a Digital Cross-Connect (DCS) is defined [AN95] [AN95a] as a device that has the ability to switch data from a given input port to a specified output port. An Optical Cross Connect (OXC) shown in Figure 2-4 is a type of DCS that interfaces with optical fiber and switches data between wavelengths or fibers. For the rest of this discussion we will use the term OXC to refer to both Digital and Optical Cross Connects. In most cases, when discussing network design, there is no difference logically whether the cross-connect is optical or digital. Like ADMs, all cross-connects have the same add-drop functionality that permits local traffic to be added or dropped. An OXC may have hundreds of fibers terminated on its interfaces. Each fiber may also support different line-rates. The primary difference between an ADM and an OXC is in the quantity of traffic that can be switched. An OXC typically has a larger switch core than an ADM. In addition, OXCs are more flexible, reliable and available. Therefore a group of ADMs cannot necessarily replace an OXC. The smallest unit size that can be switched is called the '*granularity*' of the switch. Granularity is the property of the device and is essentially dictated by the processing capabilities, buffer size and speed of the line-cards. OXCs typically support very fine granularities and may support switching between interfaces with different granularities. An OXC is therefore relatively more expensive, although the cost difference is gradually coming down given the extensive adoption of OXC technology. OXC architecture is shown in Figure 2-4. Central to any OXC is a cross-connecting switch fabric called the switch 'core'. The switching fabric in optical layer switches is non-blocking – i.e. it can connect all inputs to their corresponding outputs simultaneously. While the actual data transport is completely in the optical domain, OXCs currently still need to convert the optical signal to the electrical domain before making any routing/switching/drop decisions. In other words, the core of the switch is still electronic. These types of switches are also often referred to as Optical-Electrical-Optical (OEO) switches. Electrical processing is currently limited to about 40Gbps, whereas a fiber can carry as much as 30Tbps. It is therefore attractive to be able to switch optical wavelengths without converting into the electrical domain. All-optical or O-O-O core switches are being researched extensively; however, commercial availability of scalable O-O-O switches is still a few years away.

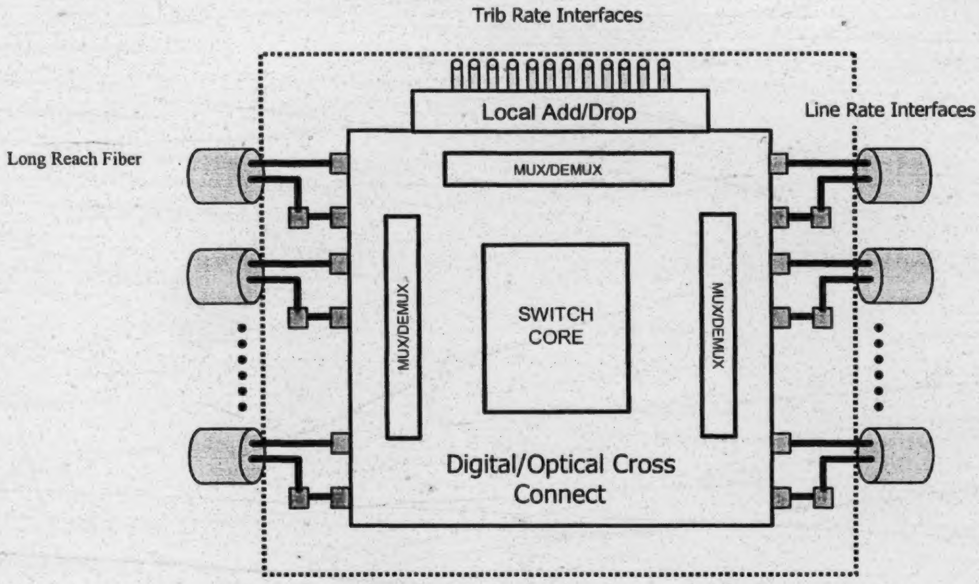


Figure 2-4. Digital/Optical Cross Connect

2.3. Technology at the Channel – SONET, GigE

A 'Span' is defined as a collection of point-to-point transmission systems between two nodes. Typically a single span is a set of cables co-routed in the same ducts. Each cable has multiple fibers, and each fiber may carry many multiplexed signals. A channel is usually the smallest granular quantity that can be switched at the end-nodes of the span. In designing transport networks, usually the number of channels on a span is termed span capacity. In most cases, it is the type and number of line-cards at the terminating nodes that defines the total number of channels that are available on any span. In this section, we will briefly examine the various technologies used to transmit data across physical spans, and define clearly, what channels mean in various contexts.

2.3.1. SONET/SDH

Synchronous Optical Network (SONET) is by far the most widely known physical networking standard. SONET owes its origin to the breakup of the AT&T Bell monopoly in the United States. All the smaller companies created as a result of the breakup started creating incompatible and proprietary equipment to run their point-to-point transport systems. The SONET [AN95][AN95a] ITU standard emerged to formalize the definitions of networking protocols and equipment used in the transport network so that all equipment such as transmitters, receivers, regenerators, etc. would interoperate even if purchased from different vendors. All SONET equipment follows the standard and relies on precise time synchronization with other nodes and therefore provides for access to direct tributary level data frames (called DS-0) without repeated demultiplexing. A SONET link is a time domain multiplexed (TDM) transmission system that multiplexes a set of data frames into the allotted time-slots. Nodes may use Global Positioning Systems or clock recovery to maintain synchronization. Demultiplexing is simplified as the end-nodes simply have to recover the data from the right time-slot. The international version of SONET called Synchronous Digital Hierarchy (SDH) essentially has similar features but is more tuned to European standard digital rates.

The significance of SONET is that it defines standard functional models for all the elements of a transport network – such as a standard DCS, standard line-cards, etc.

and defines standard acceptable error rates, survivability requirements, framing, data rates, etc. Standard digital data rates range from the DS-0 at approximately 1.5Mbps to the DS-3 at about 51.4Mbps.

Digital Signal K (DS-K) is a term for a series of standard digital transmission bitrates or levels that are multiples of the basic 64kbps DS-0 bitrate, the channel capacity required for the carriage of a single voice call. In North America, it is the T-Carrier System and in Europe, it is the E-Carrier system that operates using the DS-0 as the base granularity. DS-1 is a carrier signal in a T1 which is 24 DS-0's multiplexed using Pulse Coded Modulation (PCM) and Time Division Multiplexing (TDM). DS-2 is the concatenation of four DS-1 signals multiplexed together yielding a net bitrate of 6.312 Mbps. DS-3 is the next higher carrier signal for the T3 carrier. T3s carry 28 DS-1 signals or 672 DS-0s at a net data rate of 44.736Mbps. The ITU-TS guidelines differ somewhat from the ANSI T1.107 specifications that govern DS signals. The data rates and notations are summarized in Table 1.

Table 1. Digital Bitrates

DS-X	Data Rate	Number of DS-0s	T-Carrier	E-Carrier
DS-0	64Kbps	1	-	-
DS-1	1.544Mbps	24	T1	-
-	2.048Mbps	32	-	E1
DS-3	44.736Mbps	672	T3	-
-	139.264Mbps	2048	-	E4

The basic building block in SONET is the Synchronous Transport Signal – 1 (STS-1) frame. The STS-1 bitrate is 51.840 Mbps and can transport a single DS-3 signal. STS-1s are the channels in a SONET network. SONET manages higher data rates by byte interleaving an integral number of STS-1 frames. For example, interleaving 3 STS-1 data frames gives the standard STS-3 data rate. Some services that need a single container at an STS-3 data rate (for example) are transported by concatenating three STS-1 frames together such that there is overhead in only one frame and data payload is distributed across all three STS-1 frames. This process is called clear-channel concatenation. The resultant signal rate is termed as an STS-Nc channel, where N is the number of STS-1 channels that are concatenated. Many higher rate combinations all the way up to STS-192 may be created by concatenating lower data rate signals together. Since two SONET nodes are time synchronized, extracting a DS-0 from a DS-3 – STS-1 channel does not involve repeated de-multiplexing, instead it is a simple process of reading bits during the correct time intervals. A key point to keep in mind is that SONET does not statistically multiplex data.

The STS signal is not directly suitable for modulating the laser that eventually generates the optical signal. Continuous sequences of 'ones' or 'zeros' in the bit stream could throw off clock recovery at the intermediate nodes. Prior to transmission, the STS signal is re-formatted to ensure that the resultant signal does not go over the laser's bandwidth. Such a signal is called an Optical Carrier (OC-N) signal, where an OC-3 optical signal is a transformation of an STS-3 electrical signal. Standard SONET data rates are shown in Table 2. We refer the reader to [AN95] and [AN95a] for more information about SONET. In the next section we briefly discuss some recent advances in SONET.

Table 2. Standard SONET data rates

SONET Signal	Optical Signal	Data Rate
STS-1	OC-1	51.84 Mbps
STS-3	OC-3	155.25 Mbps
STS-12	OC-12	622.08 Mbps
STS-24	OC-24	1244.16 Mbps
STS-48	OC-48	2488.32 Mbps
STS-192	OC-192	9953 Mbps

2.3.2. Generic Framing Procedure (GFP), Virtual Concatenation (VCAT), Link Capacity Adjustment Scheme (LCAS)

A single LHN may provide transport services to a variety of virtual networks that use different protocols. With the explosive growth of Ethernet, network operators are finding that a large percentage of their payload is Ethernet data traffic. Keeping Ethernet payloads in their original protocol format is of great advantage to the network operator. This eliminates the latency caused by frequent transformations from one format to another. Additionally for packet switched communication, it is important that the data source has statistically multiplexed access to the full capacity of the SONET pipe. Statistical multiplexing, burst traffic, etc. are not supported in basic SONET. A separate problem, specific to Ethernet is caused because SONET only supports the standard data rates in Table 2. A standard Gigabit Ethernet packet requires an OC-48 worth of capacity for transport. This implies that a 2.5Gbps signal is used to transport an essentially 1 Gbps + overhead signal. GFP, VCAT and LCAS are three modifications to the current SONET standard that allow for carrying non-standard data payloads in their native format and also achieve better fill-match between Ethernet and SONET. VCAT is similar to the STS-Nc concatenation scheme except that VCAT works at less than STS-1 granularities. Using VCAT, two gigabit Ethernet packets can be multiplexed into the same OC-48, improving the fill ratio. To support non-standard bitrates, a new framing format called Generic Framing Procedure (GFP) has been developed that wraps any data rate sequence for transport across SONET equipment. Link Capacity Adjustment Scheme (LCAS) is the signaling protocol through which two SONET nodes can dynamically manage and re-size the particular Group of Virtually Concatenated channels called a VC Group (VCG). GFP, VCAT and LCAS are collectively referred to as Next Generation SONET (NGS). NGS therefore allows the SONET ADM equipment to support variable bitrates and frame sizes [BoRo02][CoMa02] and by doing that significantly extends the life of SONET as a protocol of choice for transport network. A more detailed treatment of SONET and associated technologies can be found in [Grov03],[RaKauffman02] or in the standards documents [AN95],[AN95a].

2.3.3. Gigabit Ethernet (GigE), 10 GigE

With the ubiquity of the Internet, the Ethernet LAN protocol has now become the dominant payload for all transport networks. While approaches like NGS attempt to adapt traditional payload formats like SONET STS to support LAN formats, a parallel attempt is happening in pushing the Ethernet LAN protocol to the lower physical layer. For a carrier, as discussed earlier, it makes sense to be able to use Ethernet as the transport format if 95% of the traffic is also Ethernet. Gigabit Ethernet is a native LAN format for short range transport over copper and short-reach fiber interconnects. 10GigE is the new WAN transport format for IP interconnects that aims to replace SONET. 10GigE employs a shortened and simplified version of the SONET frame header and uses the same electrical to optical signal scrambling as SONET. This is attractive as it permits use of existing SONET equipment and line-cards. In addition, since 10GigE closely matches the OC-192 data rate, a better fill match between the Ethernet protocol and SONET is achieved without the use of virtual concatenation. GigE promises to be the dominant LAN protocol in the future while 10GigE seems more promising as the corresponding WAN protocol.

2.4. Wavelength Division Multiplexing

Wavelength division multiplexing (WDM) initially was a technology that was used to drastically increase the total available span capacity between existing nodes [RaMuINFOCOM99][SPIE01][Feyn02][StBa00][ToSc02]. Optical fibers typically have two low-loss spectral center wavelength bands – in the 1300nm and 1550nm range. As long as two wavelengths being launched into the fiber are in either of these two wavelength bands and are spaced sufficiently far apart spectrally, it is possible to separate them at the destination with a properly tuned receiver. WDM exploits this ability and supports multiple wavelengths per fiber. Each wavelength is capable of carrying its own payload and is essentially independent of other wavelengths in the fiber. WDM directly enables multiplying the carrying capacity of the span without any additional cabling.

2.4.1. Coarse and Dense WDM

A transmission system capable of carrying up to six wavelengths per fiber is called a Coarse WDM (CWDM) system while Dense WDM (DWDM) systems may support many hundreds or thousands of wavelengths simultaneously. In DWDM the lasers are operated at very close frequencies. In CWDM, in contrast, up to four or six lasers operate at widely separated frequencies. CWDM technology uses International Telecommunications Union (ITU) standard 20nm spacing between the wavelengths between 1310nm to 1610nm. The Ethernet LX-4 physical layer standard is an example of a CWDM system where four wavelengths near 1310nm carry a 3.12 Gbps data stream to provide a 10Gbps aggregate end-to-end link. Cable television systems based on fiber also use CWDM where two different wavelengths are used to carry the upstream and downstream signals. Since these wavelengths are spaced fairly widely apart, looser center frequency stability is permissible and hence the cost of the CWDM system is lower. With the introduction of the ITU-T G.694.1 frequency grid in 2002, it is possible to have a DWDM system which has a frequency separation of 100GHz or about 0.8nm wavelength spacing with a standard reference frequency of 193.10THz (or 1552.52 nm wavelength.) In DWDM therefore the laser sources must have extremely narrow line-width and must maintain accurate feedback control of the laser's center frequency. An added challenge in DWDM is in maintaining equal power levels across multiple closely spread frequencies as different frequencies experience different level of attenuation. These challenges currently limit very high-capacity DWDM systems to test benches. Bell has recently reported 1000 wavelength systems on their lab benches [BELLLABS].

2.4.2. OEO Versus OOO

All optical networking, the current buzzword in transport networking literature, refers to a transport network in which all processing, regeneration, switching, routing, etc. happens completely in the optical domain. The word 'transparent' applies not just because data is transported optically end-to-end, but because the system is transparent to any electrical payload that is applied at the ends. A fully transparent or OOO network requires that a unique wavelength is assigned end-to-end along all the different fiber cables along the route of the signal. This means that the path must use the same wavelength even if it passes through multiple optical nodes. This is referred to in some literature as a pure wavelength path (PWP) network. The opposite of a PWP network is called a Virtual Wavelength Path (VWP) network that is completely opaque. In a VWP network, each virtual path may use various wavelengths along its route as at each node, the path is switched and managed in the electrical domain. At each node in this network, a device called a wavelength converter (WC) converts the data from one wavelength to another. Currently WC requires electrical processing.

As a middle path between the two extremes of pure WP or transparent networks and pure VWP or opaque networks, it is also possible to have translucent networks where the idea is to either have a relatively small set of OEO processing equipment that can perform wavelength conversion and/or regeneration, or have small regions in the network called islands of transparency that are interconnected by OEO gateways. In such a network, not every node is capable of full wavelength conversion. It is therefore possible to experience 'blocking' if a wavelength continuous path is not available end to end. But several studies have found that, in the general case, having a small pool of wavelength converters that are shared by the different wavelength paths, or having a few OEO nodes distributed in the network are sufficient to reduce blocking to negligible levels.

2.4.3. Routing and Wavelength Assignment (RWA)

In a translucent network with limited wavelength conversion, there is the possibility of blocking. The routing of working paths and the assignment of specific wavelength channels to the wavelength contiguous path segments is a non-trivial problem called the Routing and Wavelength Assignment (RWA) problem. In a SONET DCS, any input timeslot can be switched to any outgoing timeslot. In an optical network, at a node that does not have wavelength conversion, it may be impossible to switch a time-slot from one incoming wavelength to another. In addition, the required wavelength may already be used on the outgoing fiber. Thus blocking in such a network can occur for two reasons, either all the channels are used up or if the required wavelength segment is not available. RWA problems have received extensive treatment in the literature [StBa00] [RaKauffman02] [Dixi03] [Rasa97]. A detailed discussion of RWA is beyond the current scope. It is however important to understand the concept of RWA because in all the design models in this thesis we assume that the RWA problem is solved as a sub-problem after the initial capacity design and is not part of the study. Assuming that all the network nodes have full wavelength conversion capabilities simplifies network design and aids in making fair comparisons between different network architectures, without confounding issues like RWA clouding the picture.

2.5. Optical Service Channel, Digital Wrapper

In a transparent optical network, the working paths are all-optical end-to-end and therefore the end-nodes can apply any payload to that light-path. It is only the end-nodes that can/need to decode the signal or verify its integrity. But regardless of transparency, there must be some form of overhead signaling to facilitate line-monitoring, fault location, protection switching, tracing, remote operations, operations, administration and maintenance, etc. In this section we review technologies that may be used for the control plane of an all-optical network.

2.5.1. Optical Service Channel (OSC)

The control plane is a separate sub-network formed out of a designated wavelength on each span called an OSC. This wavelength uses a standard payload format and bears all the signaling information for the remaining wavelengths on the same span. Only the OSC wavelength needs to be electrically processed and read. This enables the remainder of the wavelengths to remain completely in the optical domain. This implies that the cross-connects in the network must have some degree of OEO capability to process the single OSC wavelength. [CiscONS] describes the Optical Service Channel specifications for a Cisco managed switch.

2.5.2. Digital Wrapper

The digital wrapper concept is the application of SONET-like overheads to the data payload before transmission. But unlike SONET, DW does not impose frame structure, frame timing, and overheads on the payload, but adapts itself to the payload.

For any kind of data payload, it creates a 7% overhead that takes care of all overheads. At the end-nodes, a payload signal appears to be about 93% of its actual line-rate. DW provides a SONET like abstraction for light-path networks. The network planning problem reduces to simple capacity planning. All the accumulated experience about SONET network design can, for the most part, be transparently applied to the DW optical network. More information about DW is available in [Brun00] or in the standard [802.3ae].

We will return to a detailed review of optical networking and DWDM in Chapter 4 when we discuss optical layer pre-connection and its importance. In the next section, we briefly review fundamental transport network survivability mechanisms.

2.6. Fundamentals of Network Survivability

Before we proceed into discussing survivability, let us first formalize some terminology to be used in this section. A facility route for each point-to-point link is referred to as a *span*[iii]. The corresponding end-node terminating equipment that handles the data transmitted over the span is collectively referred to as a *node*. The end-to-end traffic transmitted between two origin-destination nodes is referred to as a *demand* or *service*. Multiple demands between the same end-node pairs are referred to as a *demand-bundle*.

Transport networks originated as a collection of point-to-point copper – T1 lines or trunks that transported digitized voice traffic, the dominant traffic type in networks through most of the last ten years. At that time survivability was more or less a secondary consideration over the basic requirement that the network works. The most rudimentary survivability mechanism was to simply transmit two copies of the data for each demand on the same span on different physical transmission systems or wavelengths. The tail-end node simply examined the signal quality on both incoming routes and simply switched to backup (either manually or automatically) whenever the main or *working* transmission circuit in the span failed.

While the application of this simple technique does not guarantee survivability in case of a cable cut (since both the copies are co-routed on the same span), the underlying principle of providing redundant capacity to support working re-routing is the foundation of all survivability techniques in use today. The main research problem in survivable network design is to develop a suitable compromise between two essentially opposite aims: provide enough redundant resources such that all demands can survive failures, but minimize cost. This resource minimization is achieved by sharing backup capacity to protect against uncorrelated network failures. To illustrate, a 100% redundant survivability mechanism would be like having two phone lines for a single user. If the main or 'working' phone line is cut, the user switches to the backup phone. This is an expensive solution because even if the backup phone line is not used, it still has to be paid for. Sharing of backup resources is like two users sharing a single backup phone line. If one of the working phone lines is dead, there is a backup line available. This is essentially a 50% redundant system (assuming two continuously working phone lines and one idle backup phone line). However, this reduction in backup resources comes at a cost. If both the working phones fail simultaneously, only one of the users will get backup service. Thus with the reduction in redundancy, the level of survivability also drops. However, the probability that both phone lines fail simultaneously are low. This means that the availability of a 50% redundant system may not be significantly lower than a 100% redundant system.

While providing 100% redundancy (like the two phones per user example in the previous paragraph) was economically feasible in the copper trunk telephony network, it is prohibitively expensive in photonic networks because fiber line cards and transmission cables are much more expensive per unit than their copper counterparts. The network also has to support a lot more traffic today than prior copper-trunk networks for a lot less revenue per unit of data transported. Additionally, with improvements in technology, it is now more important to protect against cable cuts than against nodal equipment failures, which are much less frequent[iv]. It is therefore desirable to provide 100% protection against a single span failure event using the least amount of redundant capacity to support backup paths.

2.7. Physical and Logical Survivability

In the optical network, survivability considerations can be built into the lowest physical layer (an example of which is the capacity doubling described earlier) where essentially all the affected traffic on one failed fiber is re-routed into the backup fiber. Alternatively, survivability can be a consideration in the higher logical layer where these demands are routed. Physical layer survivability techniques are generally "whole-fiber" because they consider the entire failed fiber span as one quantity and consider one backup route over which all the affected traffic is re-routed. This is also necessary because at the Physical layer the network operator has no idea what is actually routed in the fiber spans. The best that can be done with purely physical techniques is to restore all the traffic on the failed fiber as a single quantity. And because these techniques are whole-fiber, generally there is no sharing of redundant resources over multiple failures. We say *generally* because *p*-cycles (to follow), originally introduced as a logical layer technology, can also be used in the physical layer to achieve some level of sharing of the physical backup fiber. Physical layer survivability mechanisms are also always pre-armed and such a survivability action is generally categorized as *protection*. Physical layer techniques also usually have protection resources dedicated to the specific working resources they protect.

A single fiber contains time domain multiplexed (TDM) data frames from several different services and each such TDM frame is called a *channel*. In SONET+DWDM, channels may be individual light-paths, which may be further used to support many TDM channels per wavelength. In the logical layer, these channel quantities are considered individually for protection/restoration. Survivability is effected one channel at a time in this layer and therefore the system has many more options than just routing all the working traffic on the cut cable along an alternate route. Using this value-added routing information in the design process can result in an increase in capacity efficiency. This of course comes at a cost in terms of the higher complexity that logical layer techniques usually have because the system may have different backup routes for different affected channels/demands. We define some more terms for logical layer techniques. A dynamic post-failure reactive mechanism that discovers and cross-connects backup paths for failed working channels is called a *restoration* mechanism. Logical layer protection mechanisms also are possible and we will come back to this when we review *p*-cycles. In *pre-planned restoration*, backup routes are found and stored at the nodes prior to failure. The post failure restoration reaction will simply be to dynamically cross-connect the necessary resources along the pre-planned backup route. These definitions are not black and white, as there are some survivability mechanisms that we will review in this Chapter that are hybrids of protection and restoration.

We now review some common survivability mechanisms that are in use in industry, and discuss the pros and cons of each technique. Of special interest are *p*-cycles, SBPP and flow-*p*-cycles, discussed in the following section.

2.8. Physical Layer Techniques

In this section we discuss the two most commonly used physical layer survivability solutions, namely Automatic Protection Switching (APS) and rings.

2.8.1. Automatic Protection Switching (APS)

APS is the simplest optical network survivability technique. It is the optical layer equivalent of the basic copper trunk network survivability technique of capacity doubling that we discussed earlier. In dedicated 1+1 APS, the same data is sent across two fibers or wavelengths. The tail-end node of this APS system simply monitors the traffic on both fibers and chooses the fiber that has the best optical signal quality [AN95]. A simple 1+1 APS system is shown in Figure 2-5. Figure 2-5 shows a single working fiber co-routed on a span with a single spare fiber. The 1+1 approach implies that the backup fiber always carries a copy of the working traffic and is dedicated to the working path it protects.

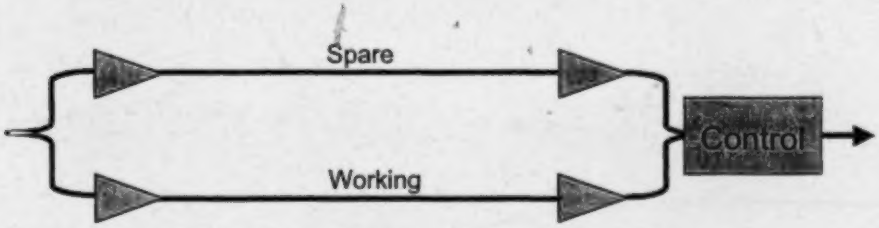
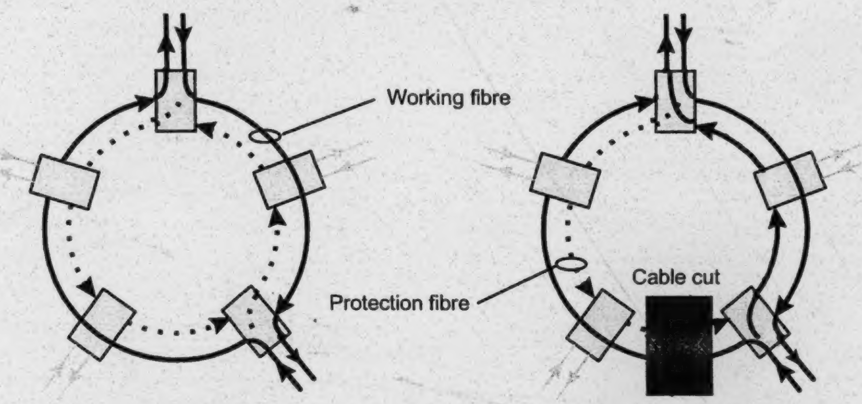


Figure 2-5. Simple 1+1 APS system. (Adapted from [DoPhD04])

In another variation called 1:1 APS, the backup fiber, when not in use for protection, routes low-priority traffic. Some providers use this to beam low-priority television signals. But the co-routing of the working and protection fibers as shown in Figure 2-5 makes this system unsuitable for protection against cable cuts. The 1+1/1:1 Diverse Path (DP) APS variation adds the requirement of geographic path diversity between the working and spare fibers, thereby ensuring survivability to a cable cut. Other variations like 1:N and M:N versions of APS allow the sharing of one (or 'M') backup fibers among 'N' working fibers, thereby reducing the total cost of the system. Routing and diversity algorithms for such systems are a separate research area in computing science and are extensively discussed in [Bhan99]. Since a single fiber many have capacities in the terabits/s range, APS is economical if a large enough demand-bundle exists between two points in the network. APS does not permit adding or dropping traffic at intermediate points, therefore it forces traffic to go all the way from origin to destination. Even in shared 1:N or M:N versions of APS (sometimes mistaken for a different logical layer scheme called Mesh Restoration (to follow)), the backup fibers are dedicated to the set of working fibers they protect, and are always between the same end-points. Using many overlapping dedicated fibers routed along many geographically diverse routes to support the ever increasing and constantly changing Internet traffic is prohibitively expensive in optical data networks. A more in-depth review of APS routing and survivability can be found in [Bhan99] and [Grov03].

2.8.2. Self-Healing Rings

Self-Healing Rings are the generation of technology that followed APS. Rings make use of capacity placed in cyclic ring physical structures [Bell95], [Bell95a]. The protection mechanism restores failures locally and allows sharing of the spare capacity only amongst the spans protected by the ring itself (i.e., links that are a part of the ring). The two main types of ring-survivability mechanisms are Unidirectional Path Switched Rings (UPSRs) and Bidirectional Line Switched Rings (BLSRs). The WDM-based equivalents are Optical Path Protection Rings (OPPRs) and Optical Shared Protection Rings (OSPRs), respectively [NeHaOFC99][MaJSAC98]. Figure 2-6 and Figure 2-7 illustrate a UPSR and BLSR, respectively. The OPPR/UPSR structure in Figure 2-6 can be thought of as a logical collection of 1+1 APS structures overlaid on a physical ring facility route. The one big advantage in UPSRs over APS is that with Add Drop Multiplexers (ADM) as nodal devices (instead of the fiber being physically patched through at intermediate nodes in APS) the operator can add or drop single channels, under the total capacity constraints of the entire ring. APS carries the entire traffic from origin to destination and is therefore justified only if sufficiently large point-to-point demand exists. This channel level add-drop capability is significant and many operators have replaced fiber patches with ADM equipment at intermediate nodes to convert their APS systems into a *folded ring* where the entire logical ring is folded onto the same physical route. There is, however, no sharing of capacity in a UPSR, as each working path on a UPSR has a working route from origin to destination on one side of the ring and an implicitly diverse backup route on the other side of the ring. This implies that a UPSR is at least 100% redundant. As shown in Figure 2-6(a), each unit demand is transmitted in each direction around the ring (solid line and dotted line) and one of the working routes survives in case of a span failure as shown in Figure 2-6(b).

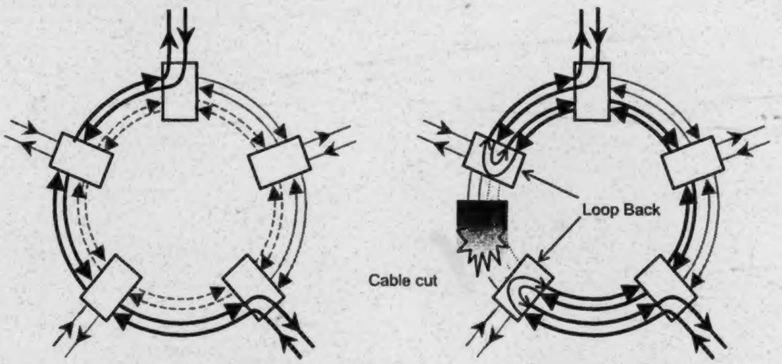


(a) Normal Operation (before failure)

b) Protection Operation (after failure)

Figure 2-6. Unidirectional Path Switched Rings. (Adapted from [Grov03])

UPSRs are essentially always two fiber dedicated structures where one clockwise fiber ring is the "working" ring with an anticlockwise "protection" ring. Bidirectional Line Switched Rings (BLSRs) shown in Figure 2-7 are like overlaid 1:N APS systems where protection capacity is shared among multiple different working sections. BLSR rings share capacity on a per-channel basis, and they can be either 2-fiber or 4-fiber structures. In 4-fiber BLSRs, as shown in Figure 2-7(a), two of the fibers are for working capacity shown by the dark lines and the other two are for backup and are shown by the dashed lines. In 2-fiber BLSRs, half the time-slots on each fiber is working capacity, while the other half is reserved for protection. In the event of a span failure, as shown in Figure 2-7(b), the end-nodes of the failed span *loop-back* the entire working traffic into the protection fibers. We can also see in Figure 2-7(a) that a demand occupies a ring only between its entry and exit points, unlike UPSRs where the demand effectively traverses the whole ring. In other words, in a BLSR, the demand does not use protection resources unless there is a failure whereas in a UPSR the protection route has to be activated at the same time as the working route. This feature permits limited re-use of capacity on the remaining spans on the ring and therefore BLSRs have slightly better traffic management capabilities. However, even with this sharing of capacity across different failures, the amount of spare capacity on any span on the ring must be at least equal to the maximum working capacity on any span on the ring to permit the BLSR loop-back reaction. This automatically forces the ring to be at least 100% redundant theoretically. If we throw in the fact that working capacities may be unevenly distributed, we can get BLSR ring networks that are up to 300% redundant in practice.



(a) Normal Operation (before failure) (b) Protection Operation (after failure)

Figure 2-7. 4-Fiber Bidirectional Line Switched Rings. (Adapted from [Grov03])

On the up side one can see how rings and APS and other physical survivability systems in general will be quite fast in restoring failed traffic. Specifically they are fast because they a) involve only the end-nodes of the failed span and b) have dedicated redundant resources for each working demand. In the best case, 1+1 APS also permits what is called 'hitless' switching where the protection switching happens without the loss of even a single bit of data. This is very much desirable in a voice network (for which rings and APS were originally designed) as even the few milliseconds that it takes to restore traffic may cause audible clicks in the otherwise clear phone conversation. Rings in general are also easy to understand and operate and require relatively inexpensive ADM nodal devices (as compared to expensive high speed cross-connect switching devices required by logical layer techniques). In metro area networks, where the low capacity efficiency of ring networks is more tolerable, rings are a simple and economical survivability solution and correspondingly rings can be expected to be around for a long time.

On the down side, in addition to the excessive redundancy requirements, rings have several significant disadvantages. Rings require the working traffic to follow the ring-constrained working path (one side of the ring), whereas the protection path is along the other side of the ring. This can be quite long and may force the placement of multiple regenerator devices to preserve end-to-end signal quality, which would otherwise be unnecessary if the working paths were allowed to follow the shortest path across the network graph. Experience has shown that while placing a single ring is a simple task, designing efficient multi-ring networks is extremely complex, and can be computationally infeasible for all but the smallest of networks. Not surprisingly, most commercial ring-design software is based on heuristics. Multi-ring networks are not easily amenable to manual re-designs, and periodic manual ring placement (to accommodate sudden growth) often results in quite inefficient real-world networks, as we shall see in Chapter 4 when we talk about network evolution. Ring networks also waste or otherwise 'strand' a significant amount of transmission system capacity (a span on a ring that has free capacity on a span that is bounded by two other spans that are saturated is said to have stranded capacity). Rings are also expensive to upgrade.

But despite these obvious problems, rings and APS were widely adopted in the 1990s because they were among the only standardized and commercially available survivability solutions at the time. Carriers all around the world have accumulated a lot of experience installing and managing ring networks and have extensive ring and APS-based networks. The work described in Chapter 4 specifically investigates options for such carriers to leverage their existing investment in ring technology to support growing network demand. We refer the reader to [Morley01] for a comprehensive overview of ring networks and ring network design. These basic ring concepts are essential to understanding *p*-cycles and how protection is generally more preferable than restoration even if there are no cost differences.

2.8.3. Recent Advances

We now list some very recent advances that have been made in Physical Layer survivability. A packet version of the BLSR ring, called the Resilient Packet Switched Ring (RPR), has recently been developed [Cisc99]. RPRs are essentially the same as Ethernet layer routers connected by fiber links that use a Spatial Re-use Protocol (SRP) (which is essentially the same as BLSR capacity re-use, except at the router level) to allow data applications to use the unused channels to improve performance during the nonfailure times. Failures are handled by a mechanism similar to BLSR loopback. RPRs allow network operators to support, to a very limited degree, features such as scheduled demands, capacity throttling, etc. RPRs also allow IP layer features such as VLAN routing that are well understood by most network administrators, in the optical layer. RPRs use the same physical layer data format as the 10GigE format which is compatible with the data format native to enterprise LAN networks; it therefore avoids a lot of software processing in the conversion from Ethernet to SONET.

GFP, VCAT and LCAS discussed in 2.3.2 are among the other notable extensions to the basic SONET transmission technology. But despite these extensions, it is difficult to enable APS and Rings to do better than logical layer techniques (in terms of capacity efficiency) that have access to much more information about what is actually routed in the fibers. While Next Generation SONET (NGS) may extend the life of existing ring infrastructure, it is difficult to see how they can easily accommodate varied future service requirements such as multi-Quality of Service, automated provisioning, scheduled demands, dual failure protected demands, etc. What seems more possible is that NGS will extend the capabilities of the physical transmission system while the logical layer techniques (in the following section) will be used to actually effect survivability thereby getting the best of both worlds. In other words, survivability will move up to the higher layers of the transport network.

2.9. Logical Layer Techniques

Logical survivability schemes, unlike system layer techniques, are not confined to physically defined static structures like rings. In a typical restorable network, there are usually a few spare channels distributed on different fiber spans. These channels are simply unassigned and held in a generic pool until a span fails. For restoration the network dynamically forms backup paths for all affected traffic out of this pool of capacity and achieves survivability objectives. In the pre-planned case, these backup routes (for each span) can be found prior to failure of any span by an offline failure-simulating process and are stored at the nodes. After the failure the backup path is formed by cross-connecting the channels along the pre-planned backup routes. Since all these logical layer restoration (and protection) techniques can find an arbitrary backup path on the mesh-like physical topology of the network, such a survivable network is usually referred to as a *mesh survivable* network. In the following section we look at two specific types of mesh survivable networks.

2.9.1. Span-Restorable Mesh Networks

Figure 2-8 shows a typical span-restorable mesh survivable network and the corresponding restoration action.

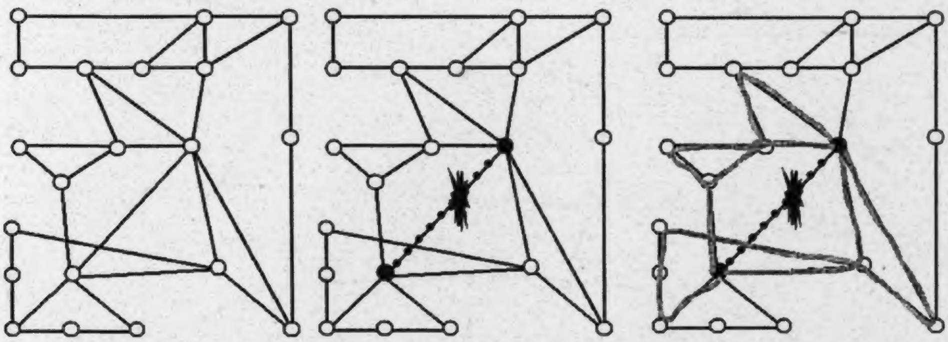


Figure 2-8. Span restoration in a mesh network (a) Mesh Topology, (b) Span failure and (c) Restoration. (Adapted from [Grov03])

The number of spans connected to a single node is referred to as the *nodal degree*. Typical transport networks have an average nodal degree between 2.5 and 6. Figure 2-8(b) shows a specific span failure for which the restoration process is shown in Figure 2-8(c). Figure 2-8(c) indicates the total restoration capacity used in the network for the multiple backup routes using the green irregular lines. Since the end-nodes of the failed span attempt to find restoration routes for the failed span, this process is termed *span restoration*. It is important to note that while span restoration considers the survivability of all the traffic on a single span as a single commodity, it may consider using multiple routes (as shown in Figure 2-8(c) using the irregular light-grey lines) to effect survivability of the many channels on a single span. A ring, on the other hand, is forced to restore all the channels along a single backup route (the surviving side of the ring.) Studies have found that the capacity efficiency of a dynamic mesh network is quite high as compared to the corresponding ring network and therefore this technique is attractive [HeByTON95]. An optimal spare capacity placement model of the path-flow type was proposed by Herzberg et al. in [HeByTON9]. The Herzberg model, as it is popularly called, is the foundation of many of the subsequent design models in this field. Span restoration in WDM networks is discussed in [KaIEEE00] with a corresponding survey of various applications of DWDM in mesh networks in [RaMuINFOCOM99]. A distributed self-organizing online mesh restoration algorithm for span restoration was proposed in [GrIEEE97][GrIEEE94]. Other researchers have studied building-in advanced capabilities like multiple survivability classes (with different levels of survivability) in an optical network in [GrClICOCN02] with an extension to support dual span failure survivability proposed in [ClGrPNET03]. While capacity is managed on a per-channel basis in span-restorable mesh networks, it can only be physically built and provisioned in standard multi-channel sized fiber modules. Typical module data rates may range from OC-3 at approximately 51 Mbit/s to OC-768 at 40 Gbit/s. There is also an economy-of-scale effect because an OC-48 module provides four times the capacity as compared to an OC-12 module but at only twice the cost. These effects are studied in [DoGrCCECE99][DoGrJSAC00]. Meta-mesh techniques that allow for efficient span restoration in very sparse networks are studied in [DoGrDRCN01] and [GrDoJSAC02]. A more detailed discussion about span restoration is found in [Grov03].

2.9.2. Path Restorable Mesh Networks

Path restoration [IrMaTON98][XiMaTON99][IrGrDRCM98] spreads the backup routing problem over the whole network by considering an end-to-end reroute of each affected working demand. Using an approach from [Grov03], we explain path restoration by pointing out the differences between span and path restoration using Figure 2-9. Figure 2-9 (a) shows a single working demand routed on a typical mesh network using an arrow tipped line. Figure 2-9(b) and (c) show the span and path restoration reactions to a specific span failure. In Figure 2-9(b), the end-nodes of the failed span use span restoration to find alternate routes between themselves, shown by the dotted arrow tipped lines. Prior capacity planning guarantees that sufficient capacity will exist in the overall network so that the failure will be restorable (more precisely, the max flow[v] between the end-nodes of the failed span in the reduced network is guaranteed to be at least equal to the number of affected working channels of that span) but there are generally no guarantees as to the precise backup routes chosen. In contrast, in path restoration shown in Figure 2-9(c), the end-nodes of the failed demand find a backup route between themselves. The capacity planning now ensures that there exists sufficient capacity in the network such that the multiple demands that fail simultaneously, because of one span failure, can be restored simultaneously. More precisely the problem model is called multi-commodity max flow, where the design process guarantees that the max flow between the end nodes of a failed demand in the reduced network will be at least equal to the number of affected working demands in each demand bundle, while simultaneously considering the restoration of all other failed demands. Pre-planned versions of both span and path restoration can guarantee the exact backup routes for a failure, but pre-planned path restoration is generally more amenable to end-user control as the entire backup route end-to-end is identified, prior to any failure. Indeed, it is only pre-planned restoration that has been used in any practical optical layer restoration mechanism. In contrast, in pre-planned span restoration one has to specify a different set of backup routes for every span failure that may affect a working path. Figure 2-9(c) also illustrates a concept in path restoration called "stub-release." In a path-restorable network it is possible to release the surviving upstream and downstream portions of a failed working path and make the freed capacity available to the dynamic path restoration process. Stub release is the main sense in which dynamic path restoration provides a *failure-specific* response. For each failure scenario there is a different envelope of stub-release capacity to be exploited. The question of stub release does not arise with span restoration because the reconfiguration that occurs is around the failed span itself. In other words, span restoration is simply a special case of path restoration with a full re-use of the stub released capacity leading up to and away from the failure as part of the end-to-end backup path. In Figure 2-9(c) we can see that the working and backup route for the demand shown in Figure 2-9(a) share a span in common at the start. The working capacity stub that was initially used in the working path for the failed demand is now re-used in the protection path of the same demand. Thus, the failure-specific stub-release action and the spreading out of the restoration routing in the network makes path restoration highly capacity-efficient. Some test networks can be contrived to do path restoration with almost no real spare capacity provided the working paths are allowed to be as long as possible.

A more in-depth review of path and span restoration and references to other recent literature on mesh network design can be found in [Grov03].

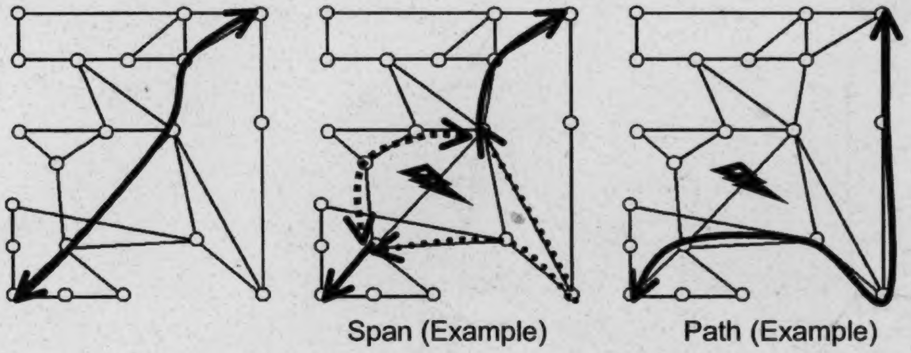


Figure 2-9. Comparison of Span and Path restoration. (Adapted from [Grov03])

2.10. *p*-Cycles

p-Cycles are a recently proposed transport network survivability scheme [StGrTON00]StGrJSAC00][GrStICC98][GrStDRCN00]. In *p*-cycle protected networks, the spare capacity is formed into cyclic pre-connected closed paths. Unlike rings, working paths are routed independently and usually along the shortest route on the network graph. *p*-Cycles are formed prior to any failure and real-time switching actions required for re-routing affected traffic are completely pre-planned and essentially similar to the BLSR rings reviewed in 2.8.2. Despite the fact that both rings and *p*-cycles use cyclic protection structures, *p*-cycles unlike rings, protect both on-cycle and straddling spans against failure. Figure 2-10 illustrates *p*-cycles protecting against on-cycle and straddling span failures. In Figure 2-10(a), a span on the cycle fails (denoted by the marked dotted line) and the surviving arc of the cycle shown using the thick irregular line is used to provide a protection path shown by the solid arrow tipped line. This switching action is functionally similar to the loop-back reaction in a BLSR ring. In Figure 2-10(b) the same *p*-cycle is accessed to support protection of a failed span that straddles the cycle.

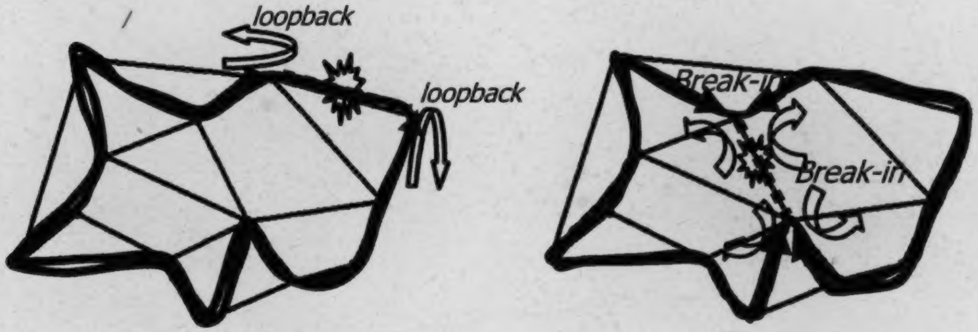


Figure 2-10. *p*-Cycle showing (a) On-cycle span failure and (b) Straddling span failure. (Adapted from [Grov03])

Straddling spans make an improvement in the failure protection provided by the available spare capacity. If a cycle is used as a BLSR, then each unit of protection capacity on the ring itself protects the same amount of working capacity, i.e. the BLSR ring needs to be at least 100% redundant. New ground is covered when the same cycle of spare capacity is used to protect spans that straddle the cycle – or in other words have both end-nodes on the cycle. For no investment of spare capacity on the straddling span itself, it is fully protected. Also seen in Figure 2-10(b), straddling spans have what is called a 2 for 1 benefit – i.e. for every unit of spare capacity on the cycle, two units of working capacity are protected on the straddling span. The resulting impact on spare capacity requirements is dramatic. In some networks, *p*-cycles can be built by construction that achieve the 1/(d-1) topological lower bound on capacity redundancy in a mesh network. *p*-Cycles can be either cross-connect based or based on ADM-like nodal devices. In the OXC based approach, *p*-cycles are formed from individual spare wavelength channels, and this offers the greatest flexibility in configuring and operating *p*-cycles. In the ADM-based network, *p*-cycles offer the same ring-like pay as you go approach to network operation. In any case, *p*-cycles do not force the working routing to be ring-like and therefore yield capacity savings that are greater than just because of straddling span protection. Working routes can either be routed along the single shortest facility route or routed jointly with the placement of *p*-cycles.

To date the main relevant advances (to *p*-cycles) in the field are:

i) **Development of an ADM-like nodal device for *p*-cycle networking**: This extended the option to implement *p*-cycle based span-protection without being cross-connect based, via a "capacity slice" nodal equipment architecture with most of the desirable cost and "pay as you grow" characteristics of BLSRs. We will review this in detail in Chapter 5 when we discuss ring-mining to *p*-cycles. (References: [StGrTON00] [GrStDRCN00] [KoGrNFOEC03]) (Patents: [StGrUS02a] [GrStUS02] [GrStCAN99])

ii) **IP-layer link-protecting *p*-cycles and node encircling *p*-cycles**: Application of *p*-cycles to IP packet layer applications for link protection and concept of node-encircling *p*-cycles (NEPCs) to provide a means for network protection against node failure. (References: [StGrJSAC00] [DoGrNFOEC03] [KaReDRCN03] [GrDoLEOS02]) (Patents: [GrDoUS03] [GrDoCAN03])

iii) **Path-segment protecting *p*-cycles**: This work, also loosely called "flow *p*-cycles" was the first extension of the *p*-cycles concept towards path-orientation. It significantly extends the ability of the scheme to include node-failure protection (without relying on separate NEPCs [DoGeDRCN05]) and it also gives a significant further increase in spare capacity efficiency over regular *p*-cycles. The main complexity of this advance is the use of path segments to deal with the mutual capacity problem[vi]. As a result, the solution that flow *p*-cycles offer is neither failure-independent nor end-to-end path-protected. Flow *p*-cycles are reviewed in detail in the following section. (References: [ShGrICC03] [ShGrJSAC03], Patents: [ShGrCAN03] [GrShUS03])

iv) ***p*-Cycle Based Protected Working Capacity Envelope (PWCE) Concept**: This most recent advance is concerned with using *p*-cycles under the "envelope" concept to support rapid, simplified, and automated provisioning of dynamically arriving and departing demand requests for protected service paths through a network. It is a bit out of scope at this point to go into PWCE in-depth, but we will come back to this in a later section where we talk about future research ideas. (References: [ShGrKLUWER04] [ShGrAPOC04])

We refer the reader to a website http://netsys.edm.trlabs.ca/p-cycles for a current list of (and cached files for) all known *p*-cycle literature and information. In the following subsection we present a detailed review of some specific topics in *p*-cycle network design that are essential background to understand the work in this thesis.

2.10.1. Flow *p*-Cycles (Adapted From [GrShUSPT] and [ShGrICC03])

The main references for this work are [ShGrICC03] [ShGrJSAC03], from which this section is adapted. These papers explain that (paraphrasing):

> *"It was natural, even at the time of the first work on span-protecting p-cycles, to ask if there was a path protection equivalent to basic span-protecting p-cycles. As simple as basic p-cycles are the latter question turned out to be difficult to address. Ultimately the difficulty is how to handle the aspect of "mutual capacity*[vi]*" contention which is intrinsic to any path-oriented or multi-commodity flow type of recovery scheme in formulating the design model under a paradigm of cyclic spare capacity structures. The corresponding operational complexity in trying to coordinate which paths can access which p-cycles, for which failures, were also beyond reach at the time."*

In flow *p*-cycles, mutual capacity problems are partly overcome by allowing the requirement of end-to-end path protection to be relaxed to become protection for arbitrary path segments. The concept improves on the spare capacity efficiency of regular *p*-cycles but even when writing the work up, the authors included recognition that the operational aspects were quite complicated, failure-specificity remained, and the simplicity of strict end to end switch-over to a predefined backup path was not achieved. The concept of flow *p*-cycles is described with the aid of Figure 2-11 (also adapted from [ShGrJSAC03] with some discussion adapted from [KoGrJLT]). Figure 2-11(a) shows a span protecting *p*-cycle that in (b) is viewed as a flow-protecting *p*-cycle. In Figure 2-11(a) the spans (0, 2), (2, 3), (3, 5), (5, 6), (6, 8), and (8, 0) are on-cycle spans of the regular *p*-cycle shown. The span (0, 5) is not on the cycle, but its two end nodes are, making it a straddling span. Note, however, that spans (6-7) and (7-2), and several others are "close to" being straddling spans but cannot actually be span-protected by the cycle shown. However, an individual service path that crosses both spans (6-7) and (7-2) as in Figure 2-11(b) can be considered to straddle the cycle shown when taking a path-level view of only the one demand that flows all the way across the *p*-cycle. Specifically the path segment (6-7-2) can be considered as a straddler to the cycle shown, between the nodes 6 and 7 as long as the total working flow remains contiguous over this segment. These basic observations lead to the concept of flow- (or segment-) protecting *p*-cycles.

Flow *p*-cycle designs can access more opportunities for spare-capacity-sharing than the span *p*-cycle method and have an additional advantage of node failure recovery. Any flow segment that intersects a flow *p*-cycle can be protected, not simply spans directly on, or straddling the cycle.

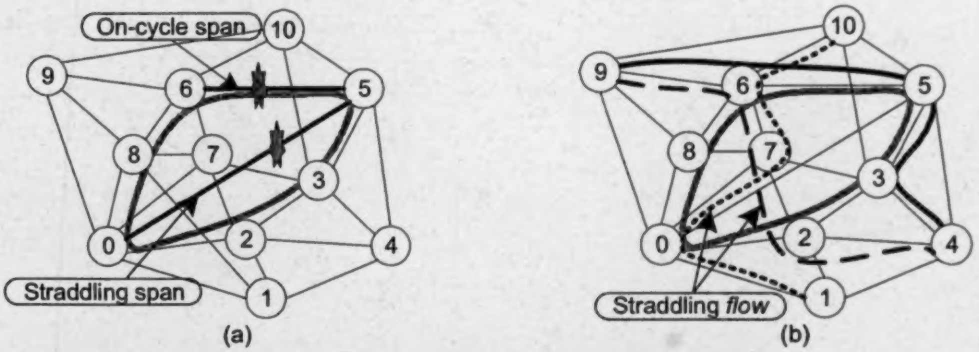


Figure 2-11. Examples of (a) a span protecting *p*-cycle and (b) the same cycle viewed as a flow-protecting *p*-cycle. (Adapted from [ShGrJSAC03])

For example, if spans (2, 7) and (6, 7) in Figure 2-11(a) incur failures, they cannot be restored by a span-protecting *p*-cycle. But, under the flow *p*-cycle shown in (b), the contiguous flows that traverse spans (2, 7) and (6, 7) can be restored by the cycle. Also, flow *p*-cycles can recover transit traffic demands due to the loss of an intermediate node on a flow. A failure of intermediate node 7 can break the flows between node pairs (1, 10) and (4, 9). The span-protecting *p*-cycle in Figure 2-11(a) cannot restore these affected transit flows. However, under the flow *p*-cycle shown in Figure 2-11(b), the flows that transit node 7 can be recovered by the cycle. Note that any demands being added or dropped at node 7 cannot be handled by the same flow *p*-cycle. The flow must be unchanged in its composition between the nodes where it intersects the flow *p*-cycle.

Given a cycle that is a candidate to be a flow *p*-cycle in a network design, the relation of any given path to the cycle can be either intersecting or non-intersecting. Only intersecting paths are relevant to the consideration of each candidate cycle. A path intersects a cycle if the two have at least two common nodes (which may include the source and destination nodes of the path). These are called intersection nodes. For example, the paths between nodes (4, 9) and (1, 10) in Figure 2-11(b) both intersect the cycle shown and are relevant to the consideration of that cycle as a possible flow *p*-cycle. By inspection, that cycle would provide straddling-type protection to the two segments (6-7-2) and (0-7-6) and on-cycle protection to an example flow such as (6-5-3) should it exist. More generally, a path can intersect a cycle in a variety of more complex ways, involving more than two intersecting nodes.

The various types of intersections between flow segments and prospective *p*-cycles have to be determined for flow *p*-cycle design. Figure 2-12(a) (also adapted from [ShGrJSAC03]) displays the simplest scenario, where the cycle and the relevant flow segment intersect at two nodes and the flow does not share any other spans or nodes with the cycle. The rest of Figure 2-12 shows other and more general intersecting flow relationships, which can be arbitrarily complex, such as Figure 2-12(e). Thus, in flow *p*-cycles a pre-processing program is used to identify all the protection relationships between candidate cycles and the corresponding flow segments.

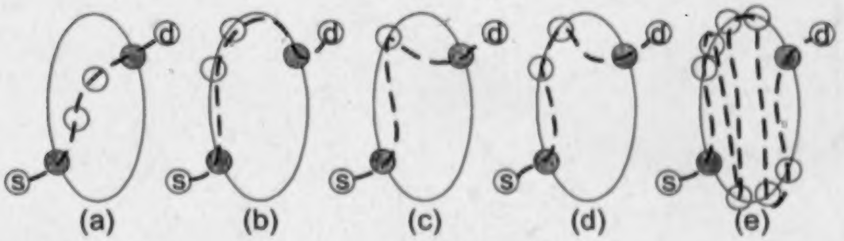


Figure 2-12. Various flow-to-cycle relationships that complicate flow *p*-cycles. (Adapted from [ShGrJSAC03])

The main new operational considerations in flow *p*-cycles (that are not needed with span *p*-cycles) are the need to locate and transmit information in real time and the need for a mechanism by which the flow *p*-cycles are employed in a failure-specific way also in real time. In flow *p*-cycles it is assumed that at the time a *p*-cycle *j* is established through a node *x*, a list of the corresponding protected flow segments that intersect the cycle at that node is recorded in association with the *p*-cycle. The Signal_ID of the working path of which the *p*-cycle protects a segment is also recorded at the node for each span failure on the flow segment. In effect, this data sets up matching conditions for node *x* to know locally which working signals (if any) it should switch into *p*-cycle *j*, depending on which span fails on any of the flow segments passing through it. Upon failure, node *x* is either adjacent to the failure, in which case it sees LoS (Loss of Signal), or the AIS (Alarm Inhibit Signal) is inserted downstream by the two nodes adjacent to the failure. All working signals bear a unique signal_ID in their overheads and, any time a node inserts AIS, it appends the ID of the incident span that has failed. Thus, the failure indication data {AIS, Signal_ID=Z, span_ID=*k*} passes through all nodes on the failed path. But only node *x* will have been "pre-wired" with the matching conditions to associate Signal_ID=Z with locally accessible *p*-cycle *j* if an indication of its failure arrives, arising from span *k*. Thus a logical matching rule can be applied at any node seeing an AIS indication to quickly determine if it has a custodial responsibility to do protection switching for the failed signal.

We describe flow *p*-cycle operation using Figure 2-13. Figure 2-13(a) illustrates a typical set of demands of which we use two - CF routed C-A-O-N-F and AB routed over A-O-M-X-Y-D-B to show flow *p*-cycle operation. The three flow *p*-cycles 1) A-O-N-X-

G-E-C-A, 2) A-M-E-G-D-Y-F-N-O-A and 3) F-D-B-H-F are illustrated using the thin differently shaded curved lines in Figure 2-13(a). Figure 2-13(b) then shows a specific span failure on span ON. Of the two demands considered – CF and AB, only demand CF is affected as it is routed C-A-O-N-F. We can see that the flow *p*-cycle #2 provides a protection path for segment C-A-O-N of demand between CF and the backup route of demand CF in the event of failure on span ON is C-A-M-E-G-D-Y-F. Figure 2-13(c) and (d) then show a different span failure on span AO. This time both demands CF and AB (routed A-O-M-X-Y-D-B) are affected simultaneously. In this scenario, flow *p*-cycles #1 and #2 are involved. Flow *p*-cycle #2 in Figure 2-13(c) provides a backup route to segment A-O-M-X-Y-D of demand A-B and the new route for the demand is A-M-E-G-D-B. Similarly flow *p*-cycle #1 provides a protection route to failed segment C-A-O-N of demand CF and the new route is C-E-G-X-N-F. But both these routes are only valid in the event that span AO fails. While these pre-armed segment protection reactions can be pre-coded in the network, it requires failure detection at each node for correct activation of the flow-protecting *p*-cycles in a failure-specific way. Since the switching reactions are failure-dependent, fairly complex pre-planning information must be established and maintained at each node for each flow *p*-cycle going through it, so that each *p*-cycle knows which failed path segments it has custodial responsibility for, depending on each possible span failure. We can see now that flow *p*-cycles are failure-specific and do not protect end-to-end paths.

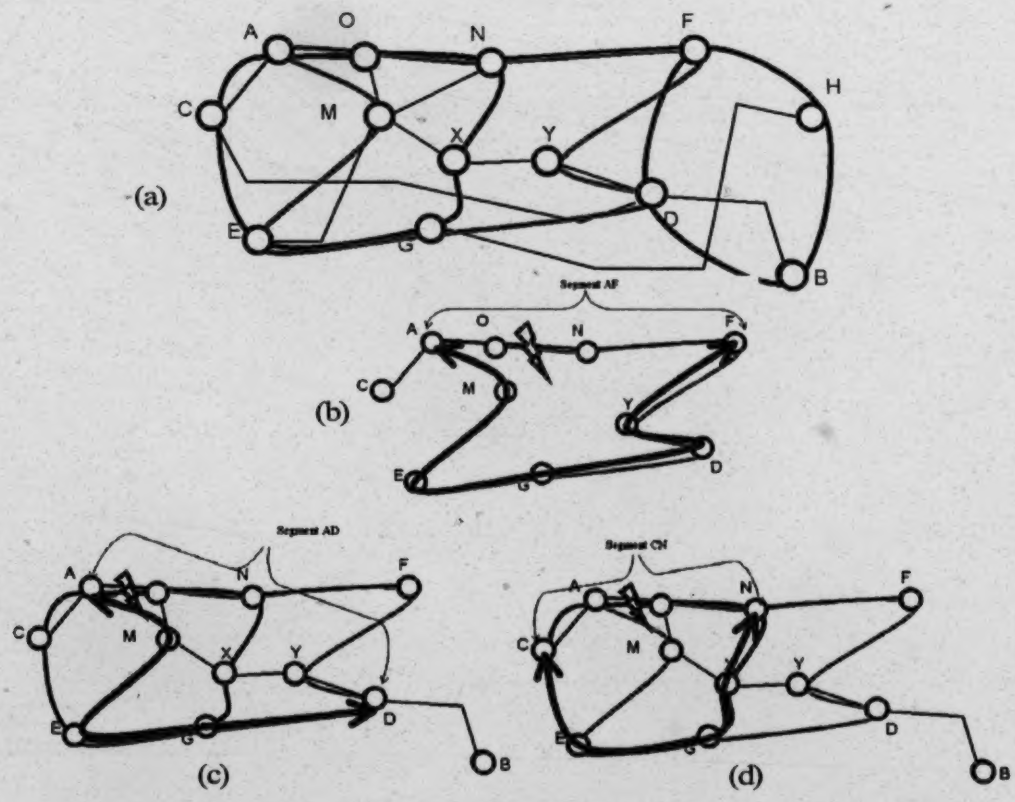


Figure 2-13. Flow-*p*-cycle operation (a) Typical set of flow-*p*-cycles, (b) Protection of segment A-O-N-F on failure of span ON, (c) Protection of segment A-O-M-X-Y-D for the failure of AO and (d) Protection of segment C-A-O-N for failure of span AO. (Adapted from [GrKoUSPT])

2.11. Shared Backup Path Protection (SBPP)

SBPP is a pre-planned path restoration scheme standardized by the IETF [KiKoIETF01] for use under Internet-style signaling protocols for protection of light-paths in optical networks. SBPP is also similar to the ATM Virtual Path concept discussed in [GrZhDRCN98]. Under SBPP one backup route is predefined for each working path and no matter what fails on the working path, restoration is via a path assembled on-demand over this one predetermined backup route. Conceptually, SBPP is like 1+1 APS DP where two fully disjoint routes, a working and a backup are established for each signal but for efficiency in the use of spare capacity, we can share spare channels over the backup routes for different working paths. For this reason SBPP is also sometimes described as 1:1 APS with "backup multiplexing." The working paths are usually called "primary" paths. Usually, one or more backup routes are possible between the same end-nodes of the primary path, and may be allowed as eligible routes to the design process, but only one is chosen for the final design. To be eligible as a backup route, it has to have no nodes or spans in common with the route of the primary path itself *and* no spans or nodes in common with any other primary path whose backup route has any spans in common with the route being considered. Together these considerations ensure that when a primary path fails (under any single failure scenario) no span or node along its backup route is simultaneously affected. This means it will be possible to assemble a backup path along that route if sufficient spare channels have been pre-planned. Figure 2-14 illustrates the concept of spare capacity sharing on predefined backup routes under SBPP.

Figure 2-14(b) shows a set of four mutually disjoint working routes between node pairs A-C, E-G, L-G, L-H. Figure 2-14(c) shows possible routes of the disjoint protection paths for the four primaries (working paths). For example, demand A-C follows the working route A-B-C for which the corresponding protection route is along A-D-O-G-C. Since these working routes are all mutually disjoint, spare capacity can be shared on their backup routes. For example, the A-D-O-G-C and E-D-O-G share the D-O-G segment and therefore as shown in Figure 2-14(d) we require only one channel of spare capacity on D-O-G per working channel on working routes A-B-C and E-P-G. The grey shaded areas in Figure 2-14(d) indicate the three spans, DO, OG and NO where sharing is possible. Of these OG achieves maximal sharing with four separate working routes sharing a single unit of spare capacity (per unit working capacity) along the backup route. Note that, in Figure 2-14, it is individual spare channels that SBPP is organized to share and that it is not possible to have these channels cross-connected in advance of failure because it is not known which of the specific backup paths in Figure 2-14(c) might be needed until the failure actually occurs. Ultimately, it is because SBPP sharing is structured on a per-channel basis over groups of mutually disjoint primaries, that SBPP requires cross-connection in real time to form actual restoration paths.

Optimization models for SBPP design are available in [DoClONM03] and are developed in more depth in [Grov03]. More often, however, heuristic methods are used for SBPP network design such as in [OuICC00] and [XiXuJLWT03]. The emphasis on heuristics for SBPP is partly because of the difficulty of solving the optimal SBPP design model even when the complete set of demands is given at once. Heuristics are also better suited for incremental survivable routing, which is a strong practical orientation for the SBPP approach. "Incremental" in this context refers to the problem of routing new demands individually as they arrive and arranging the shared-capacity backup path for each arrival in the context of all other already present demands and backup paths. Variations on this type of heuristic can either assume given capacities or try to perform survivable routing so as to minimize blocking. Alternatively each of a set of demands can be treated in sequence while attempting to route each one so that the least additional new capacity has to be added with the ultimate aim of approximating an optimal solution for collective routing and minimum total spare capacity placement problem.

The key ideas for routing under SBPP are that one tries to route the working path over the shortest or least cost path over the graph while also considering the possible backup routes (disjoint from the chosen working path) and their potential for sharing of spare capacity. On an incremental arrival basis, the optimal incremental SBPP setup is one where the sum of capacity used for the working path plus new spare channels (that have to be provided to allow for the required backup path) is minimal. It is because the

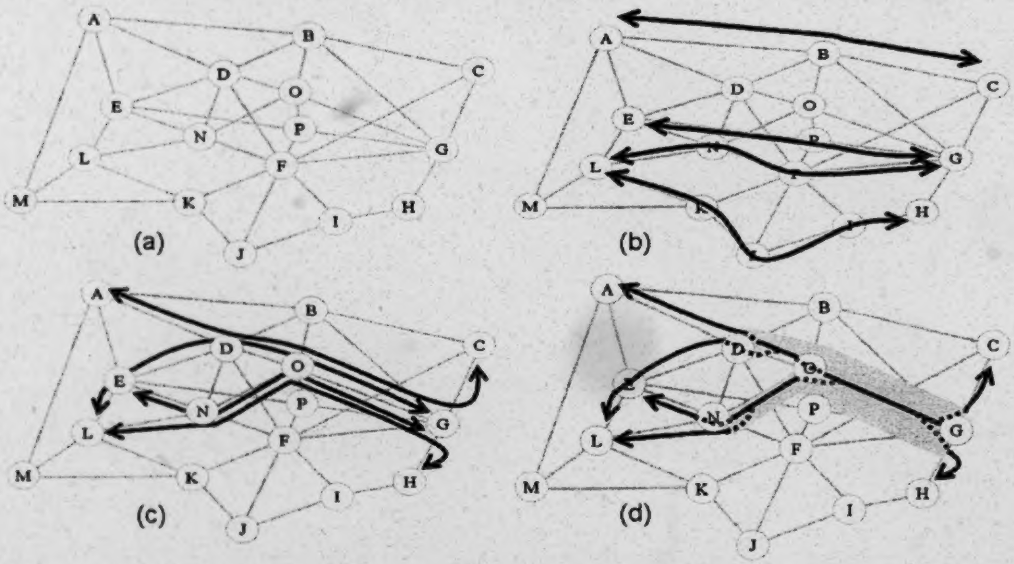


Figure 2-14. Shared Backup Path Protection illustration showing a set of four working routes that can share spare capacity (Adapted from [KoGrJLT])

backup route is fully disjoint from the route of the corresponding primary path that the switching action is "Failure-Independent." (As compared to dynamic path restoration that is failure-specific as described in section 2.9.2.) When anything goes wrong on a primary path, its end-nodes switch over to the predefined backup route. It doesn't matter what failure occurred or where it occurred along the path or whether it was a span or node failure. The existence of a failure is always immediately observable at the end-nodes and the activation and switch-over to a path on the one pre-planned backup route always follows. Under this scheme it is also possible for the users to know ahead of time exactly where their service will be rerouted in a failure. An especially important advantage of this is that fault *location* is not necessary in real time to determine the restoration response. Fault *detection* still happens in real-time, at the end-nodes, but the action to take does not depend on *localization* of the actual failure[vii]. This property is called "failure independence" and its main advantage is in transparent or translucent optical networks where fault location is slow or difficult. On the other hand, SBPP has some serious drawbacks. One of these is the need to have a coherent network state database at all times as discussed in [ShGrKLUWER04] and [ShGrAPOC04]. Each node needs to know the global capacity, topology, and backup-sharing relationships to support dynamic provisioning with SBPP. The control plane also needs a significant amount of capacity because every time a working path is provisioned or removed, or a failure occurs, the changes in the network state have to be flooded to every single node in real-time. This repeated every time a path arrives or departs itself results in a non-trivial amount of control data. There is also the issue of event horizon as path arrival and resource reservation events require a small but non-zero time interval to propagate in a reasonably sized network. In addition, SBPP depends on real-time assembly of an actual backup path, which implies signaling and length dependence of the restoration time[viii]. Such dynamic assembly of backup paths is avoided by the PXT scheme reviewed in the following section.

2.12. Pre-Cross-Connected Trails (PXTs)

PXTs are a recent proposal in [ChChTON04] with a similar concept developed in parallel in [KiLu04] that considers pre-connection of the protection path. PXTs are relevant because they use a path-oriented approach to pre-connected optical network protection. We illustrate the PXT concept using Figure 2-15 where we present a PXT solution for the same set of four working demands as shown in Figure 2-14(b). In Figure 2-14(d) one can easily see that at nodes D, O, N and G, there is no way to predict the post failure cross-connections that may be required, prior to an actual failure. In other words in the solution in Figure 2-14(d) one cannot know, for example, whether the single spare channel on span OG must be pre-connected to the corresponding spare channel on spans DO or NO at node O (and correspondingly to GH or GC on the other node G). Such nodes O and G in an SBPP solution that preclude pre-connection are defined in [ChChTON04] as branch points. The heuristic design algorithm in [ChChTON04] then seeks to specifically avoid such branching points by using three simple rules:

1) Working route is placed along single shortest route
2) Protection capacity is placed along the single shortest disjoint route and can be shared across mutually disjoint primaries.
3) Only contiguous pre-connected segments of spare capacity can be re-used and they must be used end-to-end. (no-branch-point condition)

To ensure the no-branch-point condition, the design algorithm in [ChChTON04] includes a requirement that in step 2, an existing protection path is used completely (end-to-end) or not at all. This way, the final protection structure or PXT that is formed will consist of end to end chains of protection paths and because of this, there are no branching points created. In Figure 2-15(b) therefore we see the two PXTs that are part of the final solution. PXT A-E-D-O-G-C is used for the protection of the working demand A-C routed over A-B-C. PXT L-E-D-O-G-H is used for the protection of demands E-G. L-G and L-H. Note, in Figure 2-15(b) that there are no dotted lines as shown in Figure 2-14(d). This means that there are no branch points in the solution. At nodes E and G, (and along intermediate nodes) there is no ambiguity, prior to failure, about the pre-connections to be made. Prior to failure, the two structures, A-E-D-O-G-C and L-E-D-O-G-H are fully cross-connected and in a ready-working state. But this pre-connection in PXTs comes at a price. PXTs can only ever approach SBPP capacity efficiency asymptotically (this is because every PXT solution is also a SBPP solution, while only a few SBPP solutions are PXT solutions). But the decrease in capacity efficiency may be considered a valid trade-off against all the advantages gained by path pre-connection in optical networks.

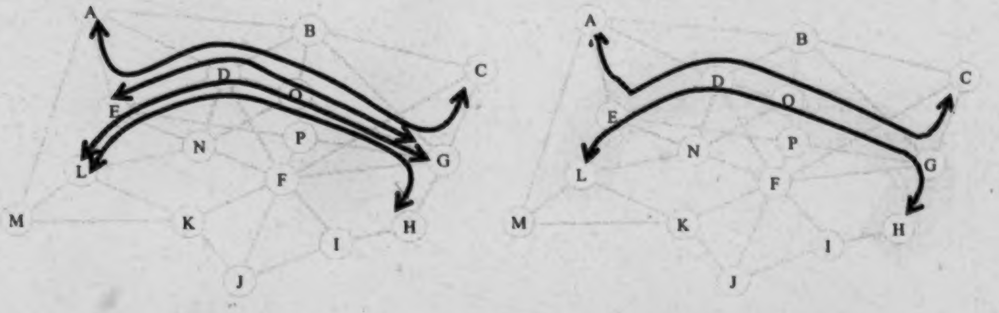


Figure 2-15. (a) PXT backup routes for working routes shown in Figure 2-14(b) and (b) Corresponding Pre-cross-connected Trails. (Adapted from [GrKoUSPT])

Given our interest in path-oriented pre-connection, we re-implemented the design heuristic in [ChChTON04] (along with a colleague Aden Grue) and found that despite our best efforts, the best network design solutions we could obtain were nearly 100% redundant and clearly not matching the numbers reported in [ChChTON04] for their own test networks. On further examination, we can see that [ChChTON04] inexplicably counters simple logic by reporting a PXT design that is more capacity-efficient than the corresponding SBPP design. The resultant PXTs from the heuristic design, are complex, long and convoluted trails that had many loops (not cycles because the looping segments are not connected at the looping nodes) and seem very complex to build. An optimal design was initially considered, but given that for a PXT-based design, one must consider as input to an optimal solver, the set of all possible pre-connection structures as input to an optimal solver, the design model quickly becomes unsolvable for all but the smallest of networks.

2.13. Demand-Wise Shared Protection

Demand-wise shared protection (DSP) is a new concept introduced by Koster et al. in [KoZyJNSM05] and [KoDRCN03] and also in a tech report [KoZyTech04]. In DSP, spare capacity is shared by the different light-paths belonging to the same demand-bundle. Each demand therefore has dedicated spare capacity. DSP combines both dedicated and spare protection. Initially the various light-paths that belong to a single demand-bundle (i.e. between the same pair of nodes in the network) are routed along spatially diverse routes. This operation, called diversification [ZyKoONDM03], distributes the working routing of the demand such that maximum connectivity between the two nodes is exploited. Each working route carries an integral number of light-paths, as sub-light-path quantities cannot be switched in a WDM context. This diversification is simply to limit the maximum outage for each demand, in the event of a failure. The innovation in DSP is to apply the diversification jointly to the routing of working and backup paths. In the most general form, the demand bundle between any two end-nodes is split over multiple mutually disjoint routes. End-to-end allocation of protection paths is also done simultaneously using maximum disjointness. Backup paths assigned to protect one working route may be co-routed with the other working routes, or separate. For instance, if three mutually disjoint routes exist between nodes A and B, and the total demand-bundle between A and B is 11 light-paths, then the best solution is to assign w_i = {0,6,5} working paths, respectively, and s_i = {6,0,0}, i=1,2,3 spare paths. In this example, the path-wise logical redundancy would be 6/(11) = 54%. The true ratio of total capacity used above that required simply for shortest path routing of demands (the "standard redundancy")[ix] would be higher than this, because in general, the disjoint paths are not shortest paths. The overall capacity efficiency is thus limited by the fact that sets of mutually disjoint routes tend necessarily to include routes that are much longer than the shortest route and by the fact that the minimum edge-cut between most node pairs in typical transport networks is only two or three. Between any two nodes with a min-cut of 2 edges, 100% is the best redundancy achievable. With a min-cut of three, it is logically 50%, but total capacity cost will not reflect this full benefit because of the excess routing lengths of the three disjoint routes involved. Ultimately these effects result in there being only 2.6% and 8.7% capacity cost reduction *relative to 1+1 APS* in the test case results for protection of all demands in [KoDRCN03]. Therefore, while the "demand-wise" shared protection scheme has some desired properties, it does not have the characteristically high-capacity efficiency that characterizes a mesh solution.[x]

2.14. Shared Risk Link Groups (SRLG) and Dual Failures

In the current context, we define three main types of failures. The first and most common kind is the single span failure. The second is the node-failure caused where some portion of the equipment at the node, or the complete node itself fails. A node failure is logically equivalent to the failure of all spans connected to it. Statistics indicate that typical LHNs experience more than one cable cut or span failure per day, and MANs correspondingly experience one every four days [FCC2][FCC3][FCC4][FCC1]. A node failure event is far less frequent but has more disastrous consequences. The third type of failure considered is a dual span failure[xi]. Typically the term "100% restorable" or the notation "R1=100" is used to imply that all the services in the network will survive a single span failure. But in such networks a dual failure can cause an outage. Dual failures can arise from a variety of reasons. In very dense or large networks standard failure statistics imply that there are bound to be a few dual failure events per year. Another cause for dual failure is a Shared Risk Link Group (SRLG) effect which is caused by unforeseen loss of route diversity between the primary and the backup path. SRLG failures occur when two spans that are different in the network are routed across some common physical facility. Some examples are when logically diverse fiber cables are routed through common ducts or pass through the same bridge in a remote area. It may be that the spans are diversely routed for all but the few feet through the bridge or duct. A service that uses one of the spans on the working route and the other span on the backup route will inevitably experience an outage when an essentially single failure event, such as a bridge collapse, or duct failure causes both the primary and backup routes to collapse. Statistical data about SRLG occurrences is not easily available but studies have been done on the effects of SRLGs on mesh network design [DoGrOPTICOMM02] [ToRaICCN04], with a corresponding recent extension to *p*-cycles in [LiLiDRCN05].

2.15. Protection vs. Restoration vs. Pre-Planning vs. Pre-Connection

System layer schemes, such as rings and APS, are inherently all protection mechanisms. Mesh restoration and *p*-cycles are generally classed as restoration mechanisms. In this section, we discuss and clarify the terms protection, restoration, pre-planning and pre-connection.

The term protection was originally used to describe the 1+1 APS system where the backup path and switching actions are completely pre-defined and in a pre-tested, ready-to-use state. The working traffic in an APS system is therefore said to be protected traffic. UPSRs and BLSRs both protect working traffic in a similar manner. In a restorable network, the main difference is that the backup paths required to restore affected traffic may have to be found and cross-connected in real-time, after the failure. Thus to characterize the difference, in a pure protection scheme, the backup paths are completely cross-connected and tested and ready to bear traffic while in a pure restoration scheme, the backup paths are simply formed, dynamically, out of the available pool of spare channels. *p*-Cycles are an example of a logical layer protection scheme while span restorable mesh networks are a logical layer restoration scheme. As an intermediate option, the backup paths and the required cross-connections to establish those backup paths for each possible span failure may be computed prior to failure and stored at the respective nodes. When the failure is detected, each node simply implements the specific cross-connection *pre-plan* it has, for the particular failure ID. This is called *pre-planned restoration*. SBPP is an example of a pre-planned path restoration scheme. If this pre-planning information is computed locally by a distributed restoration algorithm, then the mechanism is called *distributed pre-planning (DPP)* [GrMa94], [Grov97]. A logical layer scheme where the spare channels are pre-cross-connected in such a way that only the end-nodes of the failed span or light-path need to do any cross-connection is called a *pre-connected* scheme. *p*-Cycles are therefore an example of a fully pre-connected scheme. All physical layer techniques such as APS and Rings are by definition pre-connected.

2.16. Summary

In this Chapter we discussed various transport networking fundamentals. We also presented a brief review of physical and logical layer survivability. Mesh and *p*-cycle based networking was also introduced. In the following Chapter we will briefly review some basics such as graph theory, optimization and routing.

Chapter 3. GENERAL BACKGROUND REVIEW: SETS, GRAPHS, OPTIMIZATION

In survivable networking, we use various concepts and techniques borrowed from set theory, graph theory and operations research. In this Chapter we present an overview of those concepts. The aim of this Chapter is to assist the reader in developing an overall understanding of the terminology and techniques used in the rest of this thesis.

3.1. Set Theory and Related Concepts

The most fundamental concept in set theory [Krey] is the concept of a *set*, which is an unordered collection of *elements* related to each other by their very *membership* in the set. The collection of all members of a set completely specifies the set. Elements of a set may also possess some unique property solely by being members of the set. A common notation for a set S which contains elements a and b is $S=\{a,b\}$. Since sets are by default unordered, i.e. it would be equivalent to say $S=\{b,a\}$. Membership of an element a in a set S is denoted $a \in S$. To denote that an element k does not belong to set S the notation used is $k \notin S$. To refer collectively to all elements of a set, the notation used is $\forall a \in S$ where the symbol $\forall$ means *for-each*. Another quantifier called the existential quantifier is denoted by $\exists$ which means "If Exists" i.e. indicates a test to see if a specified object is a member of a given set. Thus, put together, the statement $\forall m,n \in S \times S \mid \exists k > m+n, m > n$ would mean, for any pair of elements m and n that belongs to S, for a k that is greater than the sum of m and n, such that m is greater than n.

Typically elements of a set have some defining similarity or property because of which they were selected from the population for membership of the set. An example could be the set of all real numbers R. Each element of R has the unique property that it is a real number. Thus $\sqrt{-1}$ would not be included in R. Another important distinction about sets is that duplication of members is not allowed.

The notation $S=M$ is used to indicate two sets S and M that contain the same elements. Otherwise $S \neq M$. If $M \subset N$, then all the elements of M are included in N, or M is a subset of N. Correspondingly N is called the superset of M. To indicate the possibility that *M* the subset of *N* may actually include all the elements in *N*, the notation used is $M \subseteq N$.

3.1.1. Set Operations

A union of two sets $A=\{1,2,3,4\}$ and $B=\{1,5,7\}$ is defined as a set of all elements that have membership in either *A* or *B*. The operation called $A \cup B = \{1,2,3,4,5,7\}$. The set operation called intersection is defined as a set of elements that appear in both sets. $A \cap B = \{1\}$. A complementary set operation to the intersection, called the symmetric difference ($A \vartriangle B$) is the set of elements that appear in sets *A* or *B* but not in both. $A \vartriangle B = \{2,3,4,5,7\}$. The difference between the two sets denoted by $A-B$ is the set of elements that are in set *A* but not in set *B*. $A-B=\{2,3,4\}$. A primordial set is a master set from which all elements of a set are derived. For example, the set of real numbers may be considered as a primordial set for the set of natural numbers. A complement of a set denoted by $\overline{A}$ is a set of all elements in the primordial set that are not in the set *A*. Cardinality of a set is defined as the number of elements in the set and is denoted by $|S|$. A null set is one that contains no elements. The usual symbols for null sets include $\{\}$, $\varnothing$, $|S| = 0$.

3.2. Graph Theory

A graph is an abstract mathematical concept used to represent network information in transport networking design problems. A graph $G = \{V, E\}$ is a set that contains two sets. The set $V = \{1, 2, ..., n\}$ is the set of *vertices* in multi-dimensional space and the set $E=\{1,2,3....m\}$ contains a set of *edges* that join the two vertices in multi-dimensional space. Two vertices are *adjacent* to each other if they have an edge between them. A *fully connected* graph is one where every vertice is connected directly via an edge, to every other vertice in the graph. Two-dimensional graphs can be represented on paper using geometric notation by drawing points in the plane of the paper to represent the vertices and lines to represent edges. This notation is often used to represent two-dimensional transport networks. A typical representation of a transport network is shown in Figure 3-1 where the Level 3 North American transport network is shown. When using graphs to represent transport networks, it is common use to refer to a vertex as a *node* and an edge as a *span*. A node, as we recall from 2.2 is the abstract representation of the collection of equipment such as OXCs, ADMs and Routers that are required to interface with the network. A span or a vertex is the corresponding abstraction used to represent a collection of optical fibers, regenerators and optical amplifiers that form the point-to-point transmission system.

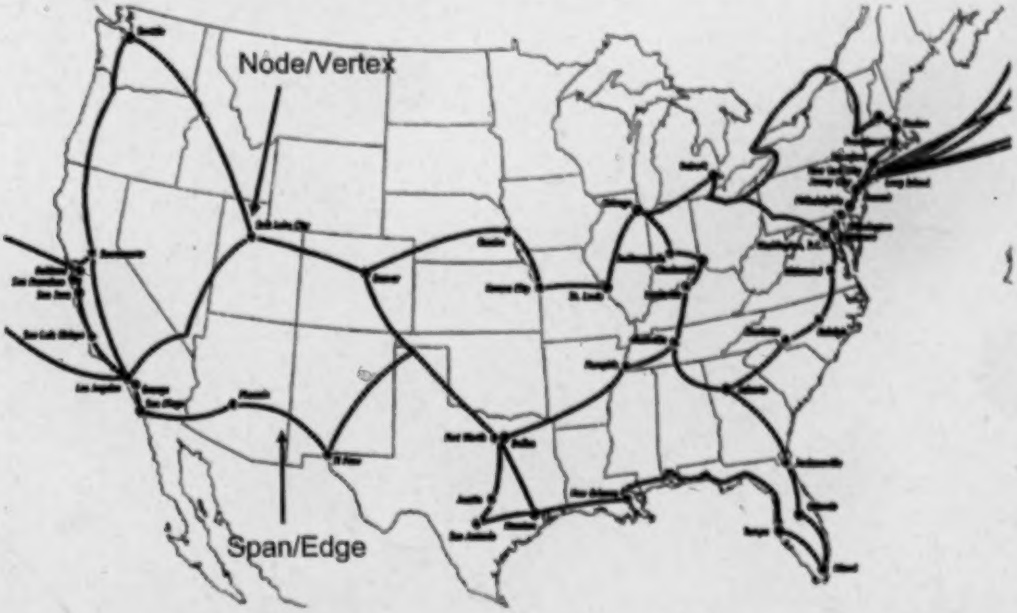


Figure 3-1. Graphical representation of the Level3 North Americal LHN. (Adapted from [Level3])

An edge with the same vertex as origin and destination is called a *self-loop*. A graph with at least one self-loop is called a *general* graph. In transport networks self-looped spans could be used for testing, optical buffering, inserting delays, etc. Two edges that are between the same pair of vertices are *parallel*. An example of parallel spans would be a 1+1 APS system without diverse path routing. Typically modern transport networks tend not to have any parallel spans as parallel spans may not provide the necessary degree of route diversity to reduce failure susceptibility to acceptable levels. A graph with at least one set of parallel edges is called a *multigraph* with the number of parallel edges between a given pair of vertices called its *multiplicity*.

If in a graph an edge has direction, i.e. for example, if the graph is representing a unidirectional fiber transmission system, then that edge is called a *directed* edge. Typical notation for directed edges is to say edge *e* = *{v1,v3}* where *{v1,v3}* is an ordered pair which implies that *v1* is the origin node of directed edge *e* and *v2* is the termination or destination. A graph with directions specified for at least one of the edges is called a *directed graph* or a *digraph*. By default all transport networks are represented as undirected graphs. Even though the fiber systems are themselves directional, there is always bidirectional communication between any two nodes connected by a span. If a span physically passes through a node, but does not actually interface with any equipment at the node, then such a node is said to have been *bypassed*. Nodal bypass is a technique that provides a low-cost physical route between two nodes in the network that have a sufficiently large demand between them but no direct physical facility route. In ring networks, any span that does not carry traffic that is added or dropped at a node may be "*glassed through*", achieving the same effect as nodal bypass.

A graph's *order* is the number of vertices in the graph and is denoted by |V|. A graph is called a *complete* graph if it is fully connected, i.e. if every vertex is adjacent to every other vertice. The number of edges incident on a particular vertex is called the *degree* of the vertex. *Nodal degree* is a very important metric in evaluating the chances of survivability of traffic incident or transiting the node. The higher the nodal degree of the network the denser or well connected, it is said to be. Vertices with degree equal to one are called *stub* nodes or *hanger* nodes. One of the fundamental requirements of survivability is that there be a possibility of finding a span and node disjoint route for the protection route (with respect to the corresponding working route), in event of a single span cut. Traffic originating or terminating at a stub node can not receive any survivability guarantees as there is no other route except through the single incident span which is a single point of failure. LHN type transport networks typically do not have stub nodes. MANs may have stub nodes, but usually this is just to carry low priority traffic that does not need a survivability guarantee. The *average nodal degree* of a graph is defined as the average degree of the vertices of the network graph. It is easily calculated as $\bar{d} = \frac{2 \cdot |E|}{|V|}$.

A graph $H = \{V_h, E_h\}$ such that $V_h \subseteq V$ and $E_h \subseteq E$ is called a sub-graph of $G = \{V, E\}$. Two graphs are called *isomorphic* if a one-to-one correspondence exists between the two graphs. An application of this property is that isomorphic graph representations of real-world transport networks are used to disguise the actual physical details of a network graph representation so as to hide sensitive information such as actual node locations and fiber facility routes. We have therefore used isomorphic graph representations of the TELUS Calgary Metro network in Chapter 5. From a transport network design point of view, isomorphic graphs are identical to the original graphs as long as the numerical information is unchanged. A *homeomorphic* graph is one where all nodes of degree two are removed and replaced with a direct span. Homeomorphic graphs are useful when studying very sparse networks that are spread out across a wide geographic area. A mesh design technique called *meta-mesh* uses homeomorphic graph representations to efficiently design mesh networks over sparse ring-like topologies.

A graph where no two edges intersect is called *planar*. Planarity is preserved across isomorphic graphs. Simple transport network graphs, as constructed in this thesis, are typically planar as fiber cables do not typically cross over one another. Of course, fiber cables may still cross over each other at close proximity to the terminating node, but that level of detail is abstracted into the node and therefore the general transport network graph still usually remains planar. In a very specific case of SRLGs as discussed in 2.14, spans may overlap for a certain segment of their route. The overall graph may still be planar. A graph with a number associated with every edge is called a *weighted* graph. The number may be used to represent capacity, cost, number of light-paths, number of fiber cables, duct size, etc. If the number is a value that represents the capacity of the span, then such a graph is called a *capacitated* graph. The smallest granular unit on a weighted edge is referred to as a *link* end-to-end while the actual capacity unit is called a *channel*. For example, in a SONET transmission system over a 4 wavelength CWDM system, each wavelength is a link/channel for the CWDM system while the individual STS-1s are the channels for the SONET system. Thus a 4-CWDM, OC-48 point-to-point transmission system has 192 STS-1 channels. Given this, we can further define spans as a set that constitutes all the working and spare channels that are on a set of transmission systems between two end-nodes. Typically all the channels on a span are routed over optical fiber cables that pass through the same duct or overhead pole. A cable cut on the span implies that all the channels through the multiple cables in a duct are cut. Industry usage of the terms span, link, channel, node, etc. is fairly confusing and sometimes link is used to refer to what we call a span here, while the word span simply means the maximum un-regenerated communication system length– i.e. a link is potentially a sequentially connected set of spans punctuated by regenerating nodes.

A *walk* through the graph is defined as an ordered set of edges that are in the order they are traversed. The first vertex of the walk is the origin while the last vertex is called the destination vertex. A walk where no edges are traversed twice is called a *trail.* A trail that has no loops (i.e. no duplicate vertices) is called a *path*. Two paths are called *span disjoint* if they don't share any edge/span in common. *Node-disjointness* is when two paths don't share any common node in their routing. Typically in transport networking, a path is a contiguous set of cross-connected channels upon which the data is transmitted. The actual facilities through with this path traverses is called a route. Thus a route is only a way through the transport network, while the path actually includes specifications such as the actual channels on the intermediate spans, port numbers, switching table entries, etc.

A trail where the origin node and the destination node are the same is called a *tour*. If a tour has no repetitions of spans or nodes, then it is called a *cycle*. A cycle is therefore simply a closed path. In the transport networking context, and for most of the work in this thesis, we will refer to both tours and cycles as just cycles. In the context of *p*-cycles we introduce two new terms. A cycle as defined here is called a *simple* cycle, while a tour that is not a cycle is called a *non-simple* cycle. Non-simple cycles may be used as *p*-cycles in specific cases if that leads to lower design costs.

A tour that traverses all the edges in a graph is called an *Eulerian tour*. The corresponding cycle is called an *Eulerian cycle*. If the cycle traverses all the nodes then it is called a *Hamiltonian* cycle. A Hamiltonian cycle is a special structure in *p*-cycle design as for some graphs, it can, by simple construction, achieve the true lower bound of span restorable network capacity efficiency. A graph is called a *connected* graph if there is at least one path between every two nodes in the graph. A graph with two span-disjoint paths between every pair of vertices is called a *two-connected* graph. A graph with two node-disjoint paths between every pair of vertices is called a *bi-connected* graph. By definition, a graph where every node is greater than degree three, is a bi-connected graph. In transport networking, two-connectedness of the network graph is typically a required property for network survivability. In other words, a working route that does not have a span disjoint protection route in the network can not survive some span failures. Figure 3-2 illustrates the three main types of graphs. Test networks used in this thesis are of type (a).

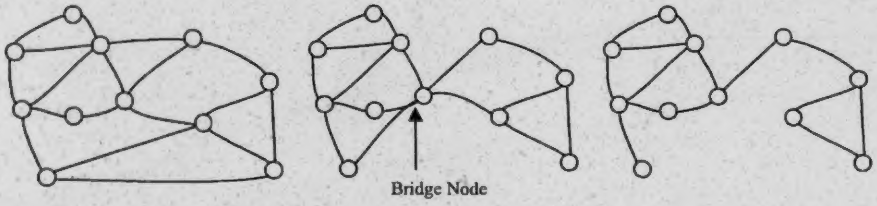


Figure 3-2. (a) a bi-connected and two-connected network, (b) a two-connected network (but not bi-connected), (c) a connected graph

A graph that is two-connected but not bi-connected usually will have *bridge nodes*. A failure of a bridge-node partitions the network into two separate components and hence disconnects the graph. Figure 3-2(b) shows a bridge-node. A stub node is a node that has degree equal to one. Figure 3-2(c) shows a network graph with a single stub node. Bi-connected graphs are also called closed graphs.

A graph that has no cycles is an *acyclic* graph. A *tree* is a connected acyclic graph. A *forest* is a disconnected acyclic graph. Each connected component of a forest is a tree. For any connected graph *G*, an acyclic subgraph *H* that contains all the vertices in G is called a *spanning tree*. A *minimum spanning tree* is one that has the shortest routes between any node pairs. A minimum spanning tree is a useful concept in data networking as it can be used to represent a collection of shortest paths between routers. If *H* were disconnected, it would be a spanning forest.

3.3. Computational Complexity

In the following sections we will discuss Linear Programming (LPs) and Integer Linear Programming (ILPs) for transport network design. It is important to be able to quantify the relative complexity of these methods as it directly corresponds to the computational time and resources required to design networks using these methods. Coming up with an exact analytical formula that denotes the complexity of an algorithm is usually non-trivial and as an effective surrogate, relative notations are used. If we can relate two functions *f* and *g* over the same domain, where *g*'s complexity is known, then it is just as useful to be able to quantize *g* as an upper or lower bound for *f* in terms of computational complexity. Asymptotic notation and *P* and *NP* classifications are used to convey the overall relative complexity of algorithms. In the following sections we briefly review these topics. For a detailed theoretical review of complexity we refer the reader to [Knuth] [CoLe01].

3.3.1. Asymptotic Notation

Suppose $f(x)$ and $g(x)$ are two functions defined on the set of Real Numbers R we can say $f(x)$ is $O(g(x))$ ($f(x)$ is big 'O' of $g(x)$) as $x \to \infty$ if and only if, there exist numbers m and n such that $|f(x)| \leq m\,|g(x)|$ for $x \geq m$. In other words – for large problem sizes, the function *f* may grow no faster than the function *g*. The function *g* in effect defines the upper bound on the complexity of function *f*. If instead $|f(x)| \geq n\,|g(x)|$ for $x \geq m$ then *f* is $\Omega(g)$ which is read as *f* is big omega of *g* and means that the function *g* definitely grows faster than *f*. *g* therefore defines the lower bound on the complexity of function *f*. Two related concepts are that of the little *o* and little ω. If *f* is $o(g)$, it means that not only is *g* an upper bound for *f* in terms of complexity, *f* is no borderline case and does not remotely approach close to the complexity of *g*. This implies that *f* is firmly within $O(g)$. The corresponding omega notation is *f* is $\omega(g)$.

3.3.2. *P* and *NP*

Asymptotic notation only conveys the worst case run times for an algorithm as problem sizes increase, in terms of another polynomial function for which the complete run-times are known. Problems for which the polynomial time algorithms exist are called Deterministic Polynomial *(P)*. An entire class of problems, called Non Deterministic Polynomial (*NP*) exists and includes *P* problems as a subset. For *NP* problems, a given solution can be verified as a valid solution in polynomial time. But there is no polynomial time algorithm to test all possible solutions. In other words, a guess can be verified in polynomial time, but there is no polynomial time method to verify all possible solutions to obtain the best one.

NP-hard problems are those that are at least as hard as any problem in *NP*. This notation can therefore apply to a problem that is not technically *NP* but its complexity approaches typical *NP* problems. Most often, this classification is because no polynomial solution algorithmic solutions have been devised for the problem. Most optimal network design Integer Linear Programming problems (to follow) are *NP-hard*. *NP-complete* problems are a special type of problem that defines an entire *NP* or *NP-hard* problem class. A solution to one *NP-complete* problem would also automatically be the solution for all the other problems in its class. A standard example is the problem of graph coloring. See [CrVi98] for a comprehensive listing of *NP* optimization problems along with approximate solution methods. In graph coloring, the objective is to assign a color to a graph vertex such that no two adjacent nodes have the same color and the minimum number of colors is used. A solution to this problem automatically provides a solution to the popular traveling salesman problem that seeks to find a minimum cycle through a network graph.

3.4. Route Finding Algorithms and Related Topics

Route finding algorithms are frequently used in transport network design to enumerate candidate routes as inputs to the design problem. The most commonly used are a family of algorithms categorized as shortest path problems. A comprehensive reference for graph algorithms is [Bhan99] with a detailed list of flow-charts in [Lau89]. In the following sections, we present a quick overview of the various algorithms used.

3.4.1. Shortest Path Dijkstra

Dijkstra's algorithm [Dijk] finds the shortest paths from a single vertex to all other vertices in a weighted, directed graph, where all weights must be non-negative. The algorithm begins by initializing any vertex in the graph (say vertex *A*) with a permanent label with the value of '*0*'. All other vertices have a temporary label with a value of '*0*'.

The algorithm then selects the least cost edge adjacent to the current vertex connecting to a vertex with a temporary label (Vertex *B* for example). Vertex *B*'s label is then updated to permanent with the value determined by summation of the preceding edge cost with the preceding vertex label value. The process is repeated iteratively, and in each iteration the temporary node with the least value is marked as permanent. The process stops when all nodes have been marked as permanent. At the end of this run, the label value of each vertice in the graph is the length of the shortest path from the origin vertex.

3.4.2. Bhandari's Variations on the Dijkstra Algorithm

Bhandari presents two variations of the generic Dijkstra algorithm. The first, called breadth-first-search (BFS), speeds up the algorithm on sparse networks with non-negative edge weights. In BFS, the algorithm simultaneously scans from all nodes that had a label value change in the previous iteration. This fan-out effect therefore quickly achieves the same effect in parallel that the original sequential algorithm had. As soon as the destination is reached for the first time, a viable route length x is known. All vertices with label values higher than x are discarded. This drastically reduces the size of the graph for the next iteration and therefore speeds up the solution. The second variation called modified-Dijkstra basically accounts for graphs with directed negative weighted edges. In the basic Dijkstra algorithm, once a node is permanently labeled, it is never re-scanned to see if there is any possible improvement. This is not a problem in undirected non-negative edge weighted graphs as there can not be any possibility of an improvement of a permanent label value. In modified-Dijkstra, all neighboring nodes of the current node are scanned. This includes nodes that have received a permanent label in a previous iteration. In effect this minor change can allow this algorithm to find a longer hop, shorter length path that uses negative edges along its path. This is particularly important in transport networks where certain edges may be allocated negative costs to signify operational details such as the cost of leasing, availability of dark fiber, etc.

3.4.3. K-Shortest Paths - All Distinct Routes

Using successive iterations of the basic Dijkstra algorithm, a tree of successive *k* shortest paths (KSP) can also be constructed. The quantity of literature on various KSP algorithms is immense, and it is beyond the scope of this thesis to survey this field. However, from the point of view of *p*-cycle and FIPP *p*-cycle network design, the two most important variations are:

***k*-shortest distinct routes**: A set of *k* successively shortest routes between a given origin and destination, where each route differs from the other route by at least one span.

***k*-shortest disjoint routes**: A set of *k* successively shortest routes between a given origin and destination, where no two routes share a common span.

Distributing working capacity across multiple span disjoint routes ensures that at least one of the routes survives a failure and a percentage of the demand is unaffected. This quick review is of particular importance when studying disjoint route sets in FIPP *p*-cycles.

3.4.4. Cycle Finding, Selection and Depth First Search

In *p*-cycle design, a standard requirement is to find good candidate cycles in the network graph to provide as input to the problem. This section describes a cycle finding algorithm that is used to generate these input data sets. Cycles are found by a simple depth first search. The search begins at a root node *k*. The next step is to begin exploring to the next unlabelled node that is adjacent to *k*, (for example *m*). *m* receives a temporary label that indicates that it is no longer available for enumeration (to prevent loops). This process continues until either the root node *k* is encountered or size limits are met. If all nodes adjacent to a current node are already labeled, then DFS simply retracts to the preceding node and proceeds to the next unlabelled node. The entire algorithm continues until there are no more unlabelled nodes adjacent to *k*. Once all adjacent nodes to *k* have been enumerated, all possible cycles through *k* have been enumerated. *k* along with all its adjacent spans, is now removed from the network graph. The search then proceeds arbitrarily to another node and this process continues until there is only one node left in the graph. In intermediate steps, if the algorithm retracts all the way to the root node, then the last seen edge is removed from the graph, as all possible cycles through that edge have been enumerated.

3.4.4.1 The Topological Score (TS) Metric

A typical transport network may have millions or even billions of possible candidate cycles. The question arises then, what is a good figure of merit (FOM) that may be used to pare down the input set? The interest in paring down the problem size is primarily due to the excessive complexity introduced by large input sets to the optimal design problem. One such FOM is called Topological Score (TS) [GrDoLEOS02]. TS is defined as:

$$TS_{i,j} = \sum_{i \in S} x_{i,j} \tag{1}$$

Where j denotes the candidate cycle in the complete universe P. $x_{i,j}$ is 0 if the span i can not be protected by cycle j;1 if span i is on cycle j;2 if span i is a straddler to cycle j. TS does not take into consideration the total capacity required to build the cycle. The logic behind TS is that that a cycle that can gather more protection relationships must be a good cycle.

3.4.4.2 The A Priori Efficiency Metric

A downside in using TS is that it always tends to favor big cycles. It can be shown that a Hamiltonian cycle has a TS = |N| + 2(|S|-|N|), which is the maximum possible value for TS in any network. An alternative metric called the a priori efficiency (AE) was proposed in [DoGrLEOS02]. The AE of a cycle j is defined as:

$$AE(j) = \frac{\sum_{i \in S} x_{i,j}}{\sum_{k \in S} \pi_{k,j} \cdot c_k} \tag{2}$$

Where $\pi_{k,j}$ is 1 if cycle j is routed over span k and c_k is the total cost of building the part of the cycle that traverses span k. A cycle with a high AE FOM is one that protects capacity more efficiently.

Both TS and AE are purely topological metrics that only indicate the *potential* efficiency of the cycle. How much of this potential is actually utilized depends completely on the actual details of the network, such as the amount of working capacity on a particular span.

3.5. Operations Research and Transport Network Design

Operations Research (OR) is the study methods of decision making under resource constraints, with the aim of optimizing factors such as cost, reliability, performance, etc. In this section we present an overview of the terminology and techniques borrowed from OR and used in transport network design. While it is important to gain a broad understanding of the tools described in this section, the tool itself is not the focus of this thesis. OR is a separate research field in itself that involves (among other things) the analysis and improvement of optimal design models. The ultimate aim in this section is to gain a user level understanding of OR so that it can be used in the study and comparison of various network architectures. Detailed information can be found in references [CaOR02][BiKeONM03] and [Winst] with a detailed survey of OR models for network design in [Grov03].

When dealing with a decision problem, the first step involves identifying all the possible variables for which values need to be obtained. These decisions are usually quantitative such as the number of widgets to manufacture or the number of light-path channels to build on a span. The second step is to identify the boundaries of the variables that help decide what values the variables can take. These boundaries are expressed as a set of mathematical statements called constraints. In the last step, one has to specify the cost benefit ratio associated with each variable – such as the total revenue from a lit light-path channel or the total cost of building a facility route for a fiber span. The set of objective function and constraints is called a data set. Mathematical programming is a body of techniques that seek to maximize or minimize the objective function subject to the constraints on the decision variables specified in the data set. Mathematical programming problems are divided into two broad classes – Linear Programming (LP) and Non-Linear Programming. Non Linear Programs allow constraints that are expressed in terms of non-linear polynomial functions. All transport networking problems dealt with in this thesis are strictly linear problems or can be simplified into piecewise linear problems. Non linear problems are therefore not discussed further.

3.5.1. Linear Programming

For a LP, the four basic elements are:

- The data set
- Sets of variables in the problem, together with their Universal Sets.
- The set of linear constraints that define the feasible solution space
- The linear function to be optimized (maximized or minimized)

We explain LP with an example (adapted from [CaOR02]) of a standard network design LP formulation called a transportation problem. Consider the rather simple road network in Figure 3-3. Each source and destination, indicated by the empty and filled in circles respectively represents grain warehouses. The objective is to ship grain from the sources 1, 2, and 3 to the destinations 1, 2 and 3 such that the needs of each destination are met. The standard LP problem is expressed as:

Parameters:

m: number of origins

n: number of destinations

u_i: amount (in truckloads) to be shipped from origin i. (where i is 1, 2 or 3)

v_i: amount (in truckloads) to be received at destination i. (where i is 1, 2 or 3)

c_{ij}: cost of sending a truckload of grain from source i to destination j. (where i,j are 1,2 or 3 and $i \neq j$)

Variables:

x_{ij}: non-negative number of truckloads to be shipped from origin i to destination j.

Objective Function:

$$\text{Min: } Z = \sum_{i=1}^{3}\sum_{j=1}^{3} c_{i,j} \cdot x_{i,j} \qquad (3)$$

Constraints:

Subject to:

$$\sum_{j=1}^{3} x_{i,j} = u_i, i \in \{1,2,3\} \tag{4}$$

(Cannot supply more than what is available at the source.)

$$\sum_{i=1}^{3} x_{i,j} = v_j, j \in \{1,2,3\} \tag{5}$$

(Don't supply more than what the destination needs.)

For the sample data set where $m=n=3$, u={2,3,4} and v = {5,2,2} the optimal solution is when the most grain is shipped at minimum cost, is Z=14, and $x_{i,j} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 2 & 0 & 2 \end{bmatrix}$.

But how does one 'solve' LPs? In 1947, George Dantzig, a mathematical advisor at the Department of Defense in the U.S. developed a solution method called the simplex method [Dantzig], that provides a solution to the above problem. A detailed description of the simplex method is beyond the current scope, and we refer the reader to [Dantzig][Winst] and [CaOR02]. As a quick overview – the simplex method is a series of sequential matrix operations that follow a few simple rules. It is eminently suitable for computational implementation and for all the LPs presented in this thesis, we use a commercial LP solver called CPLEX™ [CPLEX] developed by ILOG, Inc. We also use software called AMPL™ [AMPL] by ILOG that uses a programming language by the same name, which enables us to easily convert a basic algebraic expression of an LP model to a computer friendly format. More details about AMPL and CPLEX are beyond the scope of this review, but for now it suffices to know that these tools exist and the network planner only has to gain a general understanding of these tools to effectively design optimal capacity transport networks.

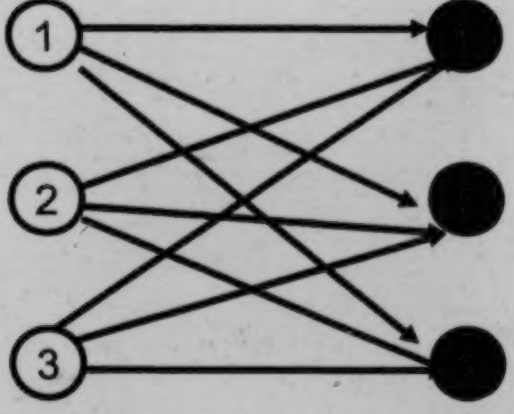


Figure 3-3. Transportation Problem Network Graph (Adapted from [CaOR02])

3.5.2. Integer Linear Programming

An integer linear programming problem (ILP) is an LP where at least one of the variables is restricted to be an integer. A mixed integer programming model (MILP/MIP) is when there is a mix of integral and real variables. A pure IP is when all variables are restricted to be integral. A 1/0 IP is a special kind of LP where at least one of the variables is restricted to be integral and only 0 or 1. In AMPL and CPLEX, one simply specifies the tag 'integer' when declaring a variable to enforce integrality. Typically transport network design problems are pure IPs, as variables such as the capacity on a span, number of regenerators, total size of a node, demand between nodes, etc. have no meaning if they are not integral. In a majority of cases, it is usual to find many 1/0 decision variables – such as: "*should module x be placed on span y?*" Pure 1/0 ILPs are extremely hard to solve and all but the smallest problem sizes are computationally infeasible to solve to optimality. In such cases, we relax, or disregard the integrality constraint on the problem or on some of the variables. This technique is called LP relaxation. For mode details on ILPs we refer the reader to [Winst] and [CaOR02]. In the following section, we briefly explain one ILP solution technique called Branch and Bound that is extensively used in solving ILPs developed in this thesis.

3.5.3. Branch and Bound Heuristic

Branch and bound is a general application of DFS to systematically search and pare down the solution space of an ILP. To apply branch and bound, one must be able to compute an upper and lower bound on the value of the objective function of the LP problem. The problem is then divided into smaller sub problems by sequential LP relaxations. The heuristic starts by solving the fully LP relaxed version of the ILP. The solution to this 'root' problem is the desired lower bound. The upper bound is simply calculated by another procedure. If the fully relaxed LP solution is integral, then an optimal solution has been found[xii]. Otherwise the feasible space is divided into two or more subsets. This is done by relaxing all but one of the variables in the problem and resolving the LP by forcing integrality on that variable. For example, if variable k is 3.4 in the root solution, the feasible space is partitioned into two sections by restricting k to be less than or equal to 3 in one region and greater than or equal to 4 in the other region. The simplex method is applied recursively in these two generated feasible space subsets. If the value of the objective function in any of the subsets is more (in case of a minimization problem) than the objective function, of the parent node, then that section of the tree and hence a chunk of the feasible solution space is cut off. The search proceeds until all variables are integer or if some specific exit criteria like run-time or MIPGAP are met.

3.5.4. MIPGAP

The MIPGAP is an allowed gap between the fully relaxed LP lower bound solution and the best currently found integer-constrained solution as a criteria for termination of the ILP run. The fully relaxed solution serves as a lower bound on the best possible feasible solution with integral decision quantities. Thus, if an ILP problem terminates when running under 1% MIPGAP, it means that the solution found is provably within 1% of the true optimum. In some types of network design problem the first feasible solution found may be close to optimal, but the fully relaxed LP version of the problem produces a very weak lower bound, so the solution is accepted when run time limits are reached even though the associated "gap" remaining to the LP lower bound may remain as much as 60 to 70%. This means that the solution could be within 1% of the optimal, but proving it would require a very long time as it has to exhaust every possible alternative. In engineering problems such as network design, it is quite acceptable if the problem is terminated prior to 1% MIPGAP if the solution generated is good enough for the specific purpose. In our experience, we find that even an 60-70% MIPGAP solution obtained to a generic ILP run on CPLEX, is far better than the corresponding custom heuristic that may take months of development time and has nothing to compare against. In any case, ILPs serve as an excellent precision microscope to examine the actual properties of network architectures without clouding the issue with other unrelated issues like heuristic performance and efficiency.

3.5.5. Benefits and Limitations of ILP-based Methods

In transport network research there is an emphasis on testing new architectures using ILP-based network design methods. Usually the models are MIPs or 1/0 MIPs. These formulations enable us to study or compare architectures based on their true optimal solutions. This eliminates any debate about the relative efficiency of the design process itself. ILPs are like an electron microscope which allows us to look into the true details of the network architecture being studied. MIP models can be used to precisely define the architecture properties. MIP can often reveal counter intuitive properties of the architecture that can lead to further insights into developing a fast heuristic. This is very much the case in FIPP *p*-cycle network design described in Chapter 8. A MIP solution even at 50 to 60% MIPGAP is often better than the best heuristic solution, while at the same time requiring only a fraction of the development time that a heuristic requires. Characterizing a network architecture using an ILP can give the network planner some idea about the range of capacity efficiency any heuristic must demonstrate to be useful.

3.5.6. Modularity and Economy of Scale

Capacity can only be installed in modules and not in single units. The capacity of the span is actually decided by the capabilities of the interfacing line-cards at the nodes. The smallest module, in a SONET network, is typically an OC-3. The next available size is an OC-12 and then an OC-48 and finally OC-192. In addition it is also usual to find that an OC-48 module costs only about twice as much as an OC-12, while at the same time provides four times the capacity. Thus, in the design problems to follow, when we make the statement about an experiment: "*Two module sizes OC-48 and OC-12 are considered with a 4x2x economy of scale*", it means that we allowed the ILP solver to choose between adding 12 or 48 units of capacity to a span. In addition, we used the pricing model where an OC-48 costs twice what an OC-12 costs but provides four times the capacity. Modularity and economy of scale are easily incorporated into ILP-based network design and we shall follow with examples in Chapters 5 to 8.

3.6. Summary

In this Chapter we reviewed the basics of graph theory, optimization and routing through a network graph. These are essential tools used for the studies in Chapters 5 to 8. In the next Chapter, we present a detailed discussion about optical networking. The next Chapter is particularly relevant background for Chapter 8 where we discuss FIPP *p*-cycles.

Chapter 4. OPTICAL NETWORKS AND PRE-CONNECTION

Optical networks are now the dominant high-capacity transmission system in the world. Fiber networks encircle the globe with over-land, underground and under-sea cables. When optical systems originated in the lab, they consisted of a laser, a length of fiber and a photodiode. Modern optical systems are far more complex and there are a large number of passive and active systems that perform complex functions to enable error-free signal transmission end-to-end at very high bitrates and distances over a wide temperature ranges.

To activate and use a live point-to-point high-capacity optical system, one needs specific skills and knowledge about the various components used in such a link. Establishing a single optical link is in itself a non-trivial task and is today a largely manual process. There are over 20 different impairments in the fiber alone, which have to be mastered before any data is actually transferred at the desired rates of 10 to 40 Gb/s and at the desired bit error rate (BER).

Ironically this factor has been almost overlooked in much of the academic research on optical network design. The tendency is to abstract these real issues in optical links to simplify design. In the literature the general assumption is that transmission systems with hundreds of WDM wavelengths may dial up end-to-end optical paths by sequentially concatenating wavelength channels (without any electrical conversion) on the fly to form working or restoration paths. The reality is that with today's state-of-the-art in switched optical networking, it is not realistic to expect that an on-the-fly concatenation of arbitrarily selected spare wavelength channels cross-connected between different spans at the optical level will result in an end-to-end path with under 10^{-15} BER. Polarization, dispersion, power levels, amplifier gain transients, and several other noise and nonlinear impairment processes must all be carefully engineered for a DWDM carrier path to achieve objectives for transmission integrity. We may hope eventually to standardize optical wavelength channel design enough, and develop adaptive power level schemes and so on, to the point where arbitrary (SONET-like) interconnection of wavelength channels into end-to-end paths is possible on demand, but this will remain difficult or expensive for some time. Without relying on OEO cross-connects to cross-connect the payloads, it remains a difficult problem to arbitrarily connect several optical channels directly with assurances of immediate end-to-end transmission quality at 10 Gb/s to 40 Gb/s in a DWDM environment.

In this Chapter we briefly review the state of the art in optical switching and networking. We then present a numerical example of how a single point-to-point 10Gbps link is designed. The relevance and the main agenda of this entire Chapter is to support our contention that on-the-fly cross-connection of optical channels to form protection paths is essentially unworkable at present. This aim is not to provide a detailed review of optical networking and equipment technologies but just to gain a basic appreciation of the technical issues involved. We wrap up with a brief discussion about pre-connection and why it is especially important in transparent optical networks. This Chapter is based on a review of a few main references: [Maxim] [Keiser] [Palais] [Hecht] and [Feyn02]. Individual references are not pointed out except in specific cases.

4.1. Optical Transmission System

A DWDM optical link can be divided into the following major sections – the Transmitter, the Optical Multiplexer, the Fiber transmission system, the De-multiplexer and finally the Receiver. A standard system diagram (Adapted from [Maxim]) is shown in Figure 4-1. In this section we briefly review the main functions of each component in this system.

4.1.1. The Transmitter

In Figure 4-1, data is first applied to the DWDM transmitter system. The DWDM transmitter system converts each input electrical signal into an optical signal. Currently DWDM systems use wavelengths in the 1530-1565nm '*L*' band or the 1570-1620nm '*C*' band. These multiple wavelength signals are then fed into an optical multiplexer (MUX). The MUX also called a signal combiner is a passive component that combines all the received wavelengths into a single multiplexed DWDM signal. Prior to transmission, the signal power of the WDM signal is either boosted using an optical booster or attenuated using an attenuation filter.

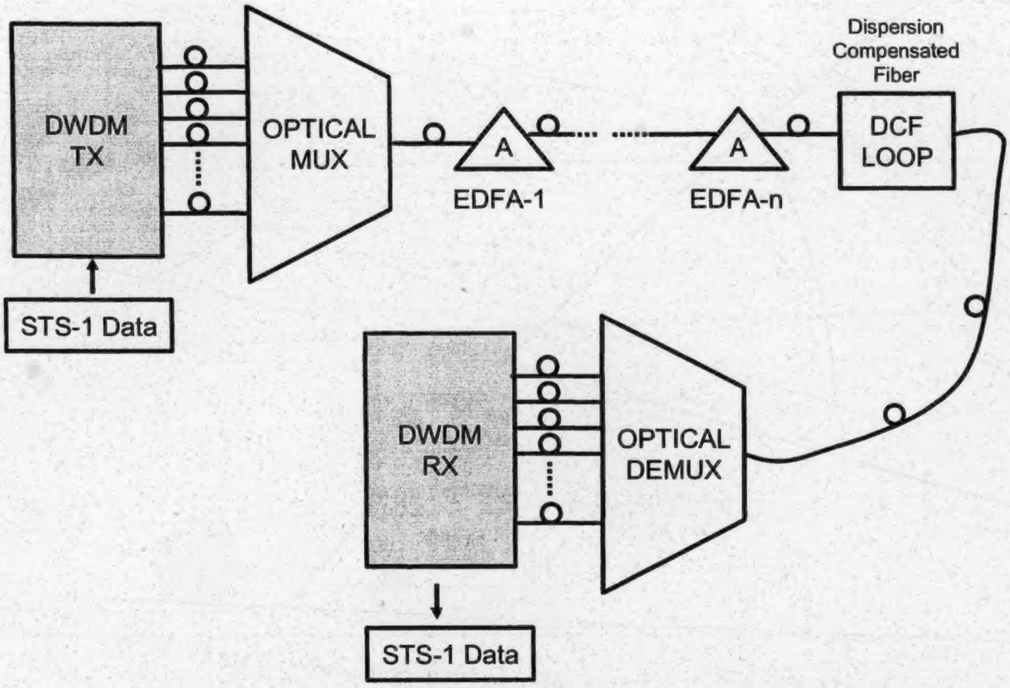


Figure 4-1. Example of a 10 Gbps point-to-point DWDM link. (Adapted from [Maxim])

Typically signal boosters (optical amplifiers) improve the reach[xiii] of the optical signal by compensating for power attenuation as the signal travels long distances. Current WDM systems achieve inter-wavelength spacing of about 0.4 nm. This works out to about 160 usable wavelengths per mux-demux combination. However, WDM transmission equipment capable of 160 wavelengths is extremely expensive, and is not in common use. The main difficulty in using these systems is at the receiver end, where separating each individual wavelength is more difficult as the inter-wavelength spacing decreases. In addition, at the transmitter end, the lasers must have a center-frequency that does not drift by even a few nm. In addition, according to our industry colleagues at Siemens Optical Networks, demand for such high capacities, such as that provided by a 160-wavelength DWDM system, does not currently exist.

Commonly used WDM systems have 4 or 20-wavelength/fiber capacities and usually do not have the ability to switch between arbitrary wavelengths. Typically only waveband level switching is supported. Currently WDM is limited to the L and C bands only, because optical line amplifier products for the 1300nm wavelength band are still not easily available.

For the transmitter, two features are extremely important. First the launched power should be such that the signal can go for long distances without the need for digital regeneration. Second the signal should have an acceptable bit error rate (under 10^{-15} BER). At the same time, the signal power should not be so high that the receiver has a non-linear response. For this reason, even if the transmitter and receiver line-cards are all purchased from the same vendor, the network operator has to carefully add attenuators or signal boosters depending on whether those line-cards are used on very short or long spans. A transmitter, fiber and receiver combination is a fairly customized triplet. If a transmitter is suddenly changed from powering one light-path to another that is shorter or longer than the light path it was originally configured for, the power level transients could easily overwhelm the receiver, or not be detected at all. In most survivable networking research on optical networking, researchers make the incorrect assumption that once the backup path is found, it can be cross-connected and 'lit' or activated in the same step. In fact, just setting up the transmission section of the end-to-end link requires a fair bit of expertise and knowledge.

There are many automatic techniques used to improve transmission. A technique that ensures a very low error rate in Optical Receivers called Forward Error Correction (FEC). While FEC is not adopted as part of standard SONET, it is implemented in some proprietary way in all WDM transmitters. In SONET over WDM systems, the FEC bytes are added to the frame overhead. Newer systems such as Digital Wrappers (discussed in 2.5.2), use standardized out-of-band FEC. In DW, the bitrate of the transmitted signal is increased to about 7% more than the serial bitrate of the payload. This overhead is where additional information, such as FEC bits, is inserted. An example of an out of band FEC algorithm is the Reed Solomon FEC algorithm defined in ITU-T G.975[xiv].

For long haul transmission systems, it is also important that the signal have very low jitter. This means that the timing jitter generated by the serializing equipment at the transmitter end and the clock-reference signals must be extremely low. To ensure low jitter, phase-locked Voltage Controlled Oscillator (VCO) based clocks are used. A standard 10Gbps DWDM transmitter architecture diagram is shown in Figure 4-2. The drive circuit must convert a voltage change electronic signal to a current change signal that is better suited to modulating lasers. The FEC encoding is done on the parallel electronic data and this is applied to a serializer to convert a low rate parallel signal into a high bitrate serial signal. A pulse width corrector, part of the laser drive circuit pre-distorts the signal based on the characteristics of the optical fiber system that follows the transmitter and effectively pre-compensates for some of the inevitable dispersion effects (to be reviewed later) that the signal will experience. In addition, at every stage, strict temperature controlling equipment is used to keep the laser's center frequencies stable. This is especially important if the transmitter system is a 160-wavelength DWDM system with very narrow (0.4nm) inter-wavelength spacing. Typically the network operator will define the fiber system characteristics in terms of a well-defined transmit power level. This power level is customized precisely to the optical path this transmitter will power. This power level is held constant by a feedback circuit, which ensures that the power level compensates for factors such as temperature change, aging, etc. Other Operations Administration and Maintenance (OAM) functions include shutdown flags, output bias current monitors and limiters, average power monitoring, etc. As a very last stage of this system, the output wavelengths are amplitude envelope modulated at a low modulation depth and frequency to enable easy channel identification.

The relevant point to take away from this section is that in an optical network, transmitters are operated in conjunction with signal boosters or attenuators and the entire transmission equipment is effectively customized (power levels, error rates, etc.) to the specific light-path it is used for. A detailed description of other blocks in Figure 4-2 is beyond the current scope.

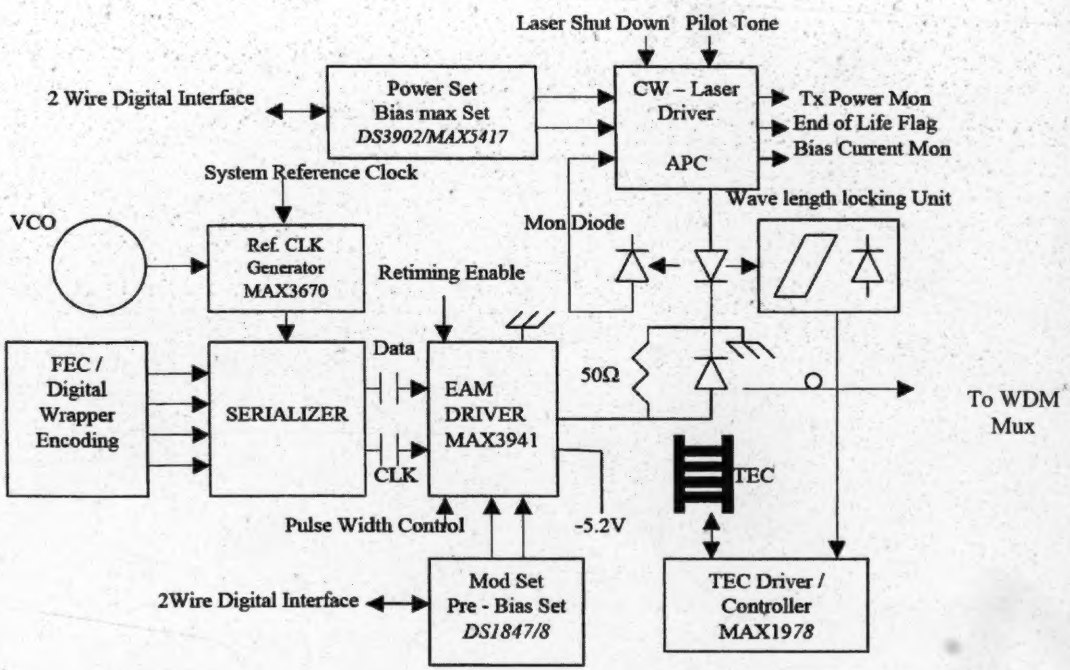

Laser Shut Down
Pilot Tone
2 Wire Digital Interface
Power Set
Bias max Set
DS3902/MAX5417
CW – Laser Driver
APC
Tx Power Mon
End of Life Flag
Bias Current Mon
System Reference Clock
Wave length locking Unit
VCO
Ref. CLK Generator MAX3670
Retiming Enable
Mon Diode
FEC / Digital Wrapper Encoding
SERIALIZER
Data
CLK
EAM DRIVER MAX3941
50Ω
To WDM Mux
TEC
Pulse Width Control
-5.2V
2Wire Digital Interface
Mod Set
Pre - Bias Set
DS1847/8
TEC Driver / Controller MAX1978

Figure 4-2. Example of a 10Gbps DWDM transmitter (Adapted from [Maxim])

4.1.2. The Receiver

Receivers can range from a simple photo detector for lab-bench experiments to complex line-cards in long haul networks that perform considerable processing and error control. Costs are also wide-ranging. Receiver architectures can be classified into two broad categories – one where all the equipment is integrated and the other where an optical transponder demultiplexes the various wavelengths in the input WDM signal to a set of output wavelengths that are then applied at the inputs of traditional single-wavelength OEO equipment. The piecemeal transponder based receiver architecture is quite popular and enables the network operator to increase the capacity of the fiber span without investing in any more physical cabling. In the integrated WDM enabled line-

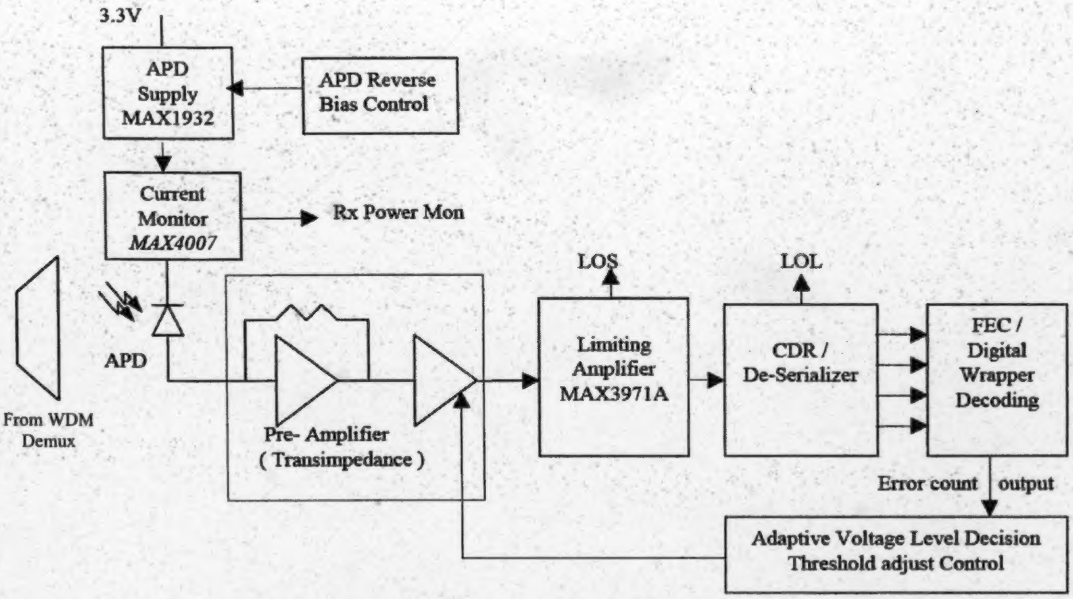

3.3V
APD Supply MAX1932
APD Reverse Bias Control
Current Monitor MAX4007
Rx Power Mon
APD
From WDM Demux
Pre- Amplifier (Transimpedance)
LOS
Limiting Amplifier MAX3971A
LOL
CDR / De-Serializer
FEC / Digital Wrapper Decoding
Error count
output
Adaptive Voltage Level Decision Threshold adjust Control

Figure 4-3. Example of a 10Gbps DWDM receiver (Adapted from [Maxim])

card, the optical demultiplexer circuit converts the incoming multiplexed wavelength signal into the corresponding individual wavelengths. DWDM systems also include components such as optical boosters, dispersion compensation fiber, pre-amplifiers, etc., all with the objective of improving the signal quality. An issue with receivers is that they have a specific dynamic power range. In other words, as the input power increases, the receiver responds linearly only over a limited range. This means that at input power levels that are much higher or lower than the receiver's upper and lower thresholds, distorted signals are received and produce erroneous digital outputs. In WDM different wavelengths experience different attenuation levels as they traverse the fiber. Prior to applying a WDM signal to a photo detector, a gain equalization circuit (analogous to a stereo equalizer) equalizes the power across all the wavelengths. Thus receivers are also customized to the input signal they expect to receive. An example of a standard 10Gbps DWDM receiver is in Figure 4-3. The input Avalanche Photo Diode (APD) is reverse biased. APDs are gain controlled by controlling their temperatures and reverse bias voltages very precisely. APDs also require a very stable ripple-free power source. A feedback controlled decision circuit distinguishes between the logic '1's and '0's and using the FEC decoding mechanism, the actual bit error rate may be calculated on the fly.

4.1.3. Amplifiers and Regenerators

Optical signals are attenuated as they travel through fiber over long distances. Below a certain power level these signals can not be reliably detected at the receiver end. Therefore amplifiers are used to boost signal power. Optical amplifiers are analog devices that amplify all the wavelengths (in a given range) applied to its input port. Regenerators are electronic digital devices that re-time, re-generate and re-transmit the signal and so are often referred to as "3R" regenerators. Regenerators require most of the same electronics as a full receiver and transmitter pair. In most cases, operators simply use a standard cross connect to regenerate the signal. For a typical transmission system, the signals are frequently amplified (typically every 80 km) using optical amplifiers along the way and every few hundred kilometers regenerators refresh the signal electronically. An all-optical regenerator is not currently available so most long haul networks are limited in reach to the maximum capacity of electrical regeneration which is between 10 and 40Gbps per light-path. Again, the amplification and regeneration is on a per-light-path basis. Therefore in a transparent or translucent network, each light-path passes through just the right number of amplifiers and regenerators that are required to reliably detect the signal at the receiver. These factors are fine-tuned when the light-path is powered on. In addition, regenerators effectively prevent payload-transparent operation as a regenerator must be able to perform functions such as re-timing and hence must know how to decode the payload format and have very specific bit timing circuits.

The Erbium Doped Fiber Amplifier (EDFA) is the most widely used optical amplifier technology. An EDFA contains a pump laser working at either 980nm or 1480nm that causes a population inversion in a section of doped fiber. Population inversion is a phenomenon where the electrons are raised to a higher energy level by applying input energy. When a low power input wavelength signal is applied to an optical amplifier, it stimulates a 'laser-like' effect. The electrons radiate the excess energy as photons and fall back into the lower energy band. These photons are the same wavelength as the input wavelength applied to the EDFA. The net effect is that the incoming optical signal is power-amplified. An optical amplifier is therefore an analog, linear technology where the complete WDM signal applied to its input port is amplified. A major limitation in EDFA technology is that they are well developed only in the lower part of the 1520-1630nm WDM wavelength range. Ongoing research is looking at other types of fiber amplifiers including those that exploit fiber nonlinearities. A significant new technology is called a Raman Amplifier (RA), after the noted physicist C.V. Raman. RAs also function like EDFAs but with the main difference that RAs do not require doped fiber sections. At the same time the disadvantage is that RAs provide very low gains. RAs can be distributed along the length of the fiber making the medium itself the amplifier. This is attractive in under-sea cables where the repair or replacement of equipment is an expensive process. Another competing technology is the Semiconductor Optical Amplifier (SOA) that is a discrete amplifier device. These are however not as well developed or as popularly used as the EDFA.

EDFAs add noise to the signal. In EDFAs the noise originates when some of the excited electrons spontaneously emit photons of random wavelengths and attain the low-energy state. If these photons are in the target wavelength range of the EDFA, this adds noise. Since a single point-to-point link may have many EDFAs chained together, and because EDFAs cannot distinguish between signal and noise, optical noise is also amplified by the subsequent EDFAs that in turn insert their own noise into the signal. This reduces the Signal-to-Noise Ratio (SNR) of the optical signal received by the receiver. This is what limits the reach of optical fiber systems and forces the network operator to include relatively expensive regenerators to refresh the signal periodically for further transmission.

4.1.4. Other Components

Connectors and splices are some of the equipment that fit into the ‘other’ category in a WDM transmission system. Connectors allow one optical fiber to be physically connected with another. Optical connectors are similar to electrical connectors but the main difference is that while electrons can follow a convoluted path through the connector, an optical connector must be clear, transparent and precisely aligned with both the coupled ends of fiber. Splices are permanent connections where the two ends of fiber are permanently joined using techniques such as fusion by heating, capillary splicing, etc. Fiber connectors and splices attenuate the signal as it transits and the number of splices and connectors along a proposed signal path are an important consideration when designing an end-to-end fiber link. When moving a laser transmitter from one light-path to another, completely in the optical domain, one has to adjust the power levels to match the differences in the attenuation characteristics of this new light-path because the number of connectors, splices and other components are different.

This completes our discussion of the main components of a DWDM system. In the next section we briefly review the effects of signal degradation when it travels through optical fiber.

4.2. Signal Degradation in an Optical Fiber

In this section, we cover two important aspects of power loss and signal attenuation in an optical fiber and noise introduced as signals travel longer distances in optical fiber.

Signal attenuation is possibly one of the two most important properties of an optical fiber. The other is dispersion. Often, it is the rate of signal attenuation that defines the maximum distances that a signal can travel, called optical reach, in the medium before being so degraded that the data carried by the signal cannot be recovered by the receiver. This therefore has a direct impact on the total cost of a fiber optic transmission system because if the reach of the optical fiber used is low, then more regenerators are needed in the network. Signal degradation therefore has two very important impacts – first it limits the capacity of the fiber itself and secondly it means that very precise system engineering is required to ensure that the point-to-point links work. In this section we develop a general understanding of the various signal degrading effects in optical fibers. We ultimately review all these considerations to support our initial contention that optical path pre-cross-connection is almost essential for workable protection in transparent optical networks.

4.2.1. Attenuation

Attenuation is defined as the loss of optical energy as the signal travels through the medium. Basic attenuation mechanisms in optical fiber are absorption, scattering and radiative losses.

Absorption is primarily a material effect and is cased by three different energy absorbing mechanisms – a) Atomic defects in the glass during fiber manufacture, b) Impurity atoms in the glass materials and c) Absorption by the constituent atoms of the fiber itself. One of the most common reasons for absorption is the presence of OH radicals from water. OH ions cause large absorption peaks at 1400, 950 and 725nm. Recall that these three wavelengths are the three low-attenuation transmission windows where optical fibers are typically used.

Scattering losses arise when there are microscopic variations in the composition and density of the fiber material. Such variations are unavoidably introduced when the fiber is manufactured. Density variations cause a variation in the refractive index of the glass material. This leads to Rayleigh type scattering (the same phenomenon that makes the sky blue.). Radiative losses are caused whenever the optical fiber undergoes a bend. Macroscopic or large bends are caused when the fiber has to be bent to physically allow routing through ducts. Microscopic bends can be manufacturing defects that are caused by non-uniformities in the process where the fiber is built into the cable. Curvature in the fiber material causes energy to leak out of the fiber. This is especially pronounced in the higher order modes in multi-mode fiber, so single mode fibers are typically used in long haul transport networks.

As a signal propagates along the fiber, in addition to attenuation, it also becomes increasingly distorted. Signal distortions are introduced as a consequence of chromatic dispersion and modal delay effects. In the next section we briefly review these two causes of signal degradation.

4.2.2. Dispersion

Low attenuation is not the only characteristic of good optical fiber. Optical fibers are extensively used because they can carry a large amount of information and at the same time have very low loss characteristics. The information capacity of an optical fiber is theoretically many terabytes per second, but in reality, dispersion effects limit the maximum information carrying capacity. Dispersion is the spreading or broadening of the optical pulse as it propagates through fiber across large distances. Dispersion effects can be broadly categorized as chromatic, intra-modal and inter-modal. Intra-modal or waveguide dispersion is a characteristic of the material of the fiber and affects both single and multi-mode fibers. Chromatic or material dispersion occurs because the refractive index of the fiber is not constant across different wavelengths. As a result, different wavelength components of the same mode may travel at different speeds and thereby lead to pulse spreading. Material dispersion is therefore important in single mode fibers. When a light pulse is launched into a fiber, it is distributed over a wide range of frequencies. Each frequency travels at slightly different speeds and therefore leads to pulse spreading.

At very high data rates, the effects of polarization mode dispersion cannot be ignored. In an ideal fiber, the cross-section is always a perfect circle. However, bending, geometric irregularities, etc. all change this cross section shape. Even the ambient temperature can affect the cross-section shape of the fiber. Signal energy of any light signal can be considered to be in two orthogonal fields. As the shape of the fiber changes, each polarization mode (field) will travel at a different velocity and generally there will be an axial rotation in the net orientation of the modes. This again causes the same pulse spreading effect. Other impairments such as mode coupling and differential mode loss in multi-mode fibers and non-linear effects also cause signal degradation but they are beyond our current scope. Dispersion effects in a fiber system are characterized as a Bandwidth Distance Product (BDP) and are a property of the fiber system. For a typical optical fiber material, BDP can be used to calculate the maximum distance a signal can be transmitted and received reliably, at a given serial bitrate.

A Dispersion Compensated Fiber (DCF) (shown at the receiver end in Figure 4-1) pre-distorts the signal at the output of the laser based on the characteristics of the light-path it is used for. One way of dispersion compensation is to insert a special kind of chirped fiber grating that serves as a frequency-selective delay line. Fiber gratings allow imposing variable delays (and corresponding pulse broadening) to different wavelengths. Therefore by adding a DCF to the transmission system, the faster moving wavelengths are delayed the most while the slowest wavelengths are let through unchanged. Thus this non-uniform delay ensures that all the different wavelengths catch up. Since dispersion accumulates over distance, a single properly designed and placed DCF can effectively cancel out all dispersion. Note however, that signal attenuation and noise are still not removed using a DCF. The basic idea is to make faster wavelengths travel longer apparent distances in the fiber. What this also means is that the DCF isn't a plug-and-play module – the transmission system has to be first characterized for dispersion effects and then a DCF module that offers just the right amount of dispersion compensation has to be designed and applied.

Thus the bit error rates in an optical transmission system are greatly dependent on the details (such as BDP) of the particular transmission system (fiber + optical amplifiers) over which the data is being sent. Typically, a network operator would need to fine tune various transmitter parameters to compensate for signal degradation effects, before being able to communicate over a fiber optic cable effectively.

In the next section, we put together all the discussion so far on the topic of optical networking and work through a practical example of designing a point-to-point optical link. Without going into detailed numerical calculations, we present a quick walk through the process any network planner must go through when commissioning a single point-to-point link. To re-iterate, our overall agenda is to argue how unlikely it is that on-the-fly concatenation of transport optical channels will result in a working end-to-end optical path.

4.3. Engineering of a Single Point-to-point Optical Link

Optical path engineering can be best described as a multi-variable balancing act. The designer starts with a set of requirements for the link and translates this into detailed engineering specifications. Requirements may be that the link support 10Gbps through a 10km fiber cable. Other requirements may be that the system be minimum cost or maximum reliability. Maximum allowable bit error rate is usually 1 error in 10^{15} transmitted bits and reliability requirements state that the system must operate continuously for 5 years. A network designer will then ask the client how much each requirement is worth. If the reliability requirement is paramount, it may make sense to spend more on high-end transmission and receiving equipment and metal sheathed optical fiber cable buried in ducts 2 meters below the ground. A brief list of common questions any designer may ask is given below:

- What is the output power of the source laser?
- What are the coupling losses between transmitter and fiber?
- What is the center frequency and line-width of the transmitter laser; or how many wavelengths can I pump through this system?
- What is the rise time and frequency response of the transmitter laser?
- What electronic wrapper format is used? Is it SONET or DW?
- How many splices and connectors along the fiber path chosen for this link?
- Are you using single or multi-mode fiber? What is the numerical aperture (or core diameter) of the fiber?
- What is the end-to-end attenuation and dispersion?
- What wavelength do you want me to operate on – or – do you want to use DWDM/CWDM or “non coloured” optics?
- In case of WDM, what are the insertion losses of the mux/demux?
- How many EDFAs exist along the fiber route?
- What is the precise signal modulation format applied to the transmitter laser?
- What kind of switching do you want to do – do you want to switch frequently at nodes along the route or is it a dedicated end-to-end light-path that I am designing?
- What is the sensitivity of the receiver?
- What is the current BER and SNR? (if the light-path is currently lit for other purposes)

- What is the bandwidth and dynamic power range of the receiver?

Designers looking at deploying inter-continental optical networks have to look at a lot more information (and investment), which sometimes can border on being completely arbitrary, such as ensuring that the power line accompanying the under-sea cable is sufficiently shielded to ward off shark bites on the cable. For our purposes it is sufficient to develop a high level understanding of the complexity of designing a single point-to-point link using current networking equipment. In the following sections we walk through some of the main points in engineering this system.

4.3.1. Power or Link Budget

Power budgets are very much like fiscal budgets in that the aims are common. A power budget ensures that enough power is launched into the circuit such that all intermediate losses and attenuation effects are properly compensated for. As a simple mathematical statement, the power budget is expressed as

$$\Omega_{\mathtt{power}} = \text{Safety\_Margin} + \text{Receiver Power Sensitivity} - \text{amplifier gain} + \text{system loss.} \tag{6}$$

All losses discussed in 4.2 must be accounted for.

4.3.2. Selection of Fiber and Light Source

Choice of fiber affects the choice of the light source. If the choice is a large-core-diameter step index fiber, and it is just an intra-office fiber connection, an LED source is a possible choice. Long haul transport on single mode fiber usually requires a Laser source.

4.3.3. Receiver Power Sensitivity

The receiver has some fairly rigid requirements. The easy part is in coupling the light from the fiber to the receiver photodiode. The hard part is in calculating the minimum received power, minimum SNR, and resulting BER, etc. also taking into account clock recovery jitter and pulse shaping filter insertion losses. Getting a receiver to accurately and reliably de-code the signal is a non-trivial undertaking and involves careful tweaking of a variety of parameters.

4.3.4. Losses Due To Connectors and Splices

The biggest problem with having many connectors in a fiber transmission system is that the losses are variable. A connector, if properly installed is characterized with a maximum possible loss factor. However, each connector is installed slightly differently and may therefore exhibit different individual dB loss. Using the worst case numbers creates an over-powered system while using the best case or average scenarios may imply higher BER and mode leakage losses.

4.3.5. Amplifiers

Amplifiers can be used to frequently increase the signal power but they are very expensive devices to use and are not available in the first low-loss window of 1300nm. In addition they also add their own noise and amplify the background noise thereby reducing the SNR of the signal. Amplifiers are relatively expensive and make economic sense only in WDM systems where their costs can be amortized over the tens of wavelengths that they simultaneously amplify. Amplifiers can also easily be overloaded by a single high-power carrier causing nonlinear distortion to all wavelengths going through the amplifier.

4.3.6. Safety Margin

A safety margin in a budget is to account for any unexpected system changes such as changes in the ambient temperature leading to higher losses, maintenance, on-line fault detection, etc. Typical system margins are about 10 to 12 dB.

4.3.7. Example Power Budget Calculation

This example is adapted from an illustration in [Hecht] (pp. 415-417) and simply serves as a numerical example of how the process works. We use the same numbers to ease calculation while significantly simplifying the problem description from [Hecht].

4.3.7.1 Problem Data

Construct a long-haul point-to-point link with the following specifications:

Length: 300km

Splices: 1/10km

Single mode fiber, 0.25dB/km loss at 1550nm

Carries a single wavelength at 10Gbps

Transmitter source is a screened DFB laser with narrow line-width (0.01nm)

Receiver is a photodetector with very high receiver sensitivity (usually -32dBm at 10^{-15} BER is a typical sensitivity spec.) This is necessary because the fiber system is a very high data rate system.

4.3.7.2 Problem Statement

Design a minimum cost 10Gbps point-to-point link.

4.3.7.3 Design

In other words, if all the existing components are connected properly, will the system work? Exact intermediate calculations are not shown – just the end numerical results.

Laser power: 0.0dBm

Fiber loss: -75dB

Splice loss: -2.9dB

Connectors: -3.2dB

Power at Receiver: -81.1dBm

Receiver Sensitivity: -32dBm

Net System Power Budget: -49.1 dB (deficit)

So it is clear – just from the point of view of the power budget that this system will not work by just using the existing components. There is a power deficit at the receiver side. Obviously the fiber itself is what adds the most losses. To compensate for this, there are three possible alternatives – a) significantly boost the power coupling between the transmitter laser, b) boost the power output of the laser and choose an ultra-low loss fiber or c) use amplifiers. Other alternatives exist, but let us say we are restricted to these three.

Option (a) may not be feasible as the laser may be operating at peak power output (1mW) already. In addition, excess power launched into the fiber may cause non-linear effects.) Option (b) is usually not available as one has to work with the fiber available and the use of a different quality of fiber may not be possible. The last option of adding amplifiers seems to be the best among the available options. So let us say that we add two amplifiers along the path, each with a gain of 30dB. (If the peak amplifier output is too high, then instead of adding two amplifiers, we may add three with less than 30dB gain. This is just an example.)

The system power budget now is as below:

Laser power: 0.0dBm

Fiber Loss: -75dB

Splice loss: -2.9dB

Connectors: -3.2dB

Amplifiers: 60dB

Power at Receiver: -81-1dBm

Receiver Sensitivity: -32dBm

Net System Power Budget: 7.9 dB (reasonable safety margin.)

Thus this system is adequately provisioned to support the power required to setup an end-to-end link.

4.3.8. Dispersion Budget

The bitrate of a digital system is intricately dependent on the dispersion in the fiber. The maximum bitrate a given fiber system can support is specified in the BDP as described in 4.2.2. So in calculating whether the system can actually support the necessary bitrate, one assumes that the system power budget has already been properly calculated. So now let us calculate whether the 10Gbps system in the previous example will actually work (in theory at least). We proceed, assuming NRZ coding, with the same problem details as in 4.3.7.1. Again the numerical values are adapted from [Hecht](pp. 422)

The fiber is a dispersion shifted fiber (DSF) with maximum material dispersion below 3ps/nm.km. The laser is a screened DFB laser with a line-width of 0.01nm. Total Dispersion is 9ps. A 10Gbps signal operates at a pulse width of about 100ps. This system therefore has a comfortable dispersion margin for operation at 10Gbps.

4.3.9. Network-level Challenges

Even if the power and dispersion budgets are properly calculated for the link, there are also network-level challenges to making the link functional. The most important challenge for any network operator is interoperability. The operator must be able to mix and match optical networking gear on his network. In addition, carriers must be able to switch light-paths between networks that belong to multiple operators. This is essential if the carrier wants to provision end-to-end light-paths where one or both of the ends are located in other carrier's networks. Typically when provisioning light-path service between Edmonton and Winnipeg, the light-path originates in the TELUS network and terminates in the Manitoba telephone system network. An example is shown in Figure 4-4. To ensure interoperability, the optical networking community is working on standardizing the User Network Interface (UNI), the Internal Network to Network Interface (I-NNI) and the external NNI so that the interface between any two components that need to talk to each other for light-path setup is standardized. Interoperability ensures that any new innovations in the network design mechanisms can be easily implemented without being confounded in the details of each vendor's equipment or networking protocol. In addition, there is a significant cost reduction as the operators can competitively procure network carriage on other networks without having to perform electrical processing. Network management therefore becomes much simpler.

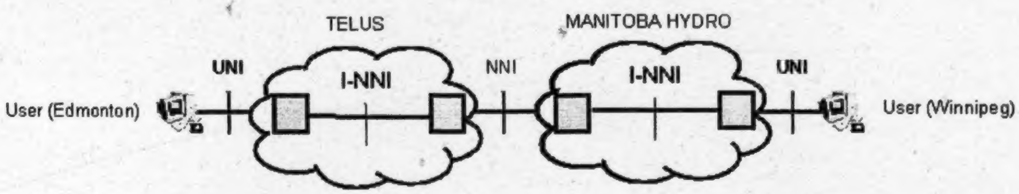


Figure 4-4. UNI, ONI, NNI for interoperability in optical networks.

In current networks, interoperability is still a critical issue that is being addressed. In other words, even if the system is engineered within the power and dispersion budgets, network-level problems such as incompatibility between the various line-cards used, interfacing between the various networks through which the light-path may pass may need to be addressed before the light-path will actually work.

4.4. Observations

As mentioned, the intent in section 4.1 to 4.3 was to illustrate how complex fiber systems are to setup and operate. To setup a single 10Gbps point-to-point link, one must carefully engineer system parameters such as dispersion, power, amplifier gain transients, inter-modal distortion, and several other noise and nonlinear impairment processes. In addition, for multi-carrier networking, one must ensure that the various pieces of equipment at the ends and along the way must all interoperate to guarantee an end-to-end working light-path. The entire process is currently largely manual and switching itself is done primarily using an electronic core cross-connect (OEO).

Consequently all-optical dynamic switching of transport wavelength channels is a topic of academic research but is not yet feasible in practical implementations. Currently research is in progress that hopes to standardize optical wavelength channel design, and develop adaptive power level schemes, and so on, to the point where dynamic and arbitrary interconnection of wavelengths is possible. Without relying on OEO cross-connects to cross-connect the payloads themselves, it is difficult to arbitrarily connect several optical channels directly with assurances of immediate end-to-end transmission quality in a DWDM environment. The real point of this exercise is to convince the reader that full pre-cross-connection of protection paths is a very important and attractive property because the backup light-paths can be manually engineered and tested prior to failure. These known-working light-paths can then be accessed for restoration. It is for this additional reason that we are motivated in this thesis to study *p*-cycles, which are an optically pre-connected, capacity efficient survivability mechanism. In the following sections we briefly review prior work on optical layer pre-connection and its advantages.

4.5. Optical Layer Pre-Connection

We discussed pre-connection briefly in 2.15. In this section we describe some recent research on the topic and explain why pre-connection may actually bring All-Optical networks closer to being practical. In the recent paper on PXTs [ChChTON04] it was observed that (paraphrasing) "*p*-cycles are fast not because they are cycles, but because the protection paths they provide are fully pre-connected before failure." This reiterates one of the original aims of the work in [StGrTON00] and [StGrTR99] and adds a renewed emphasis on pre-connection as a paramount property of interest in an optical network. Pre-cross-connected linear path segments (concatenated chain of light-path channels) or trails were initially studied in [StGrTON00][GrMaEL94][StGrTR99] and [StMSC97] where it became clear that cycles were inherently more capacity efficient than any acyclic protection structure because they can provide up to twice the number of protection relationships per unit of spare capacity. An important difference in motivation in [ChChTON04] relative to [StGrTR99] was that Chudak et al. sought a path-oriented model. The basic concept behind PXTs is the same as in that of pre-connected segments except the intent is to break into the PXTs that are present to replace failed working paths on an end-to-end basis. The pre-connection property is also a primary motivation in recent work by Shah-heydari et al. on pre-connected trees as protection structures [ShYaPNET04]. Thus, pre-connection itself is not a new topic. It was extensively studied for linear segments, trees, and arbitrary patterns including cycles as far back as 1997. But the renewed general interest in placing fully pre-connected structures of spare capacity for network protection, has highlighted the importance and desirability of *p*-cycles and of FIPP *p*-cycles.

Regardless of the efficiencies of the various pre-connected schemes, pre-connection itself is a critical property in optical networks. In an optical network, as we saw earlier in this Chapter, engineering an end-to-end optical link is a manual and fairly time-intensive process. Pre-connected paths have guaranteed optical integrity as they can be pre-engineered and tested and be in a known-working condition prior to their use. Therefore pre-connection may be one of the enabling mechanisms of implementing an all-optical survivable network.

4.6. Summary

In this section, we have reviewed optical networking and associated technologies. We also worked through an example of how a single 10G point-to-point link is set up and activated. We did this to support two motivating hypotheses of this work: 1) On-the-fly cross-connection of transport optical wavelength channels to form protection paths is fraught with many difficulties, and that hence 2) the property of full pre-cross-connection of protection paths is important, if not essential to realize survivable optical transport networks.

This concludes our review of relevant background and we now proceed to the next section where we discuss the various studies done on *p*-cycle concepts as part of this thesis.

Chapter 5. RING MINING TO *P*-CYCLES AS A TARGET ARCHITECTURE

5.1. Introduction

Migration from existing ring-based networks towards a future mesh-based architecture and operation is of considerable interest to network operators. In "ring mining" [ClGrDRCN01] the line capacity and high speed interfaces of existing ring transport systems are reclaimed to support new growth by converting to mesh-based routing and protection operating under the span capacities of the prior rings. Recent studies [ClGrDRCN01] have found high potential to support ongoing growth without new capacity additions because ring mining (i) reclaims 100% protection capacity, (ii) unlocks stranded ring working capacity, and optionally (iii) frees working paths from ring-constrained routes. So far, however, ring mining has assumed span restorable mesh as the target architecture. In this work we consider *p*-cycles as the possible target architecture. *p*-Cycles are an obvious candidate for ring mining because like rings, they too are cycle-oriented. As discussed in 2.10, *p*-cycles operate with BLSR-like switching simplicity using either modified ADMs or cross-connects, but share protection capacity around the circumference of a cycle over both ring-like on-cycle failures and mesh-like "straddling span" failures. The resultant networks are based on ring-like, predefined switching structures but are mesh-like in capacity efficiency. We present results for studies conducted on real metro networks of up-to 22 rings.

The work in this Chapter is the result of a collaborative study between TRLabs (Adil Kodian, Wayne D. Grover) and the Technology Strategy Group at TELUS Communications (Jim Slevinsky, David Moore). The bulk of this study was published recently at NFOEC 2003 [KoGrNFOEC03]. Parts of this work are also adapted into Chapter 10 of [Grov03].

5.1.1. Motivation and Goals

Throughout the 1990s carriers installed many SONET ring based transport networks to serve traditional traffic sources. With the growth of the Internet, data traffic has already overtaken voice traffic without a corresponding relationship in revenue. For example, British Telecom estimates that 80% of its revenue comes from voice which uses only 10% of its network capacity [ART03]. Data traffic growth is expected to be above 100% per annum, driven by services like voice over IP, video on demand, etc. This makes transport capacity efficiency of greater importance. Multiple quality of service, multiple quality of protection are new paradigms that are difficult to implement with ring networks. At the same time experience has accumulated showing that while a single ring is very simple, multi-ring networks are extremely hard to design, operate and grow, especially if demand is difficult to forecast accurately. To serve an OC-192 worth of demand on one ring span, a service provider may have to upgrade all other nodes of the ring to OC-192 capability– even if not needed by the traffic on the other spans. For these and other reasons carriers are increasingly looking towards more efficient and flexible mesh-based networks as the way to go in their future transport planning. In this Chapter, we present a study on migrating rings to *p*-cycles.

5.1.2. Prior Work on Ring to Mesh Migration

The question is, therefore: "How to get from an existing network of rings towards a mesh-based future network?" One approach is to freeze the legacy ring network and serve growth in a new mesh network from scratch. But it is of interest to see if we can be more efficient with existing assets than this baseline of a "cap and grow" strategy. Severi and Wellbrock [SeWeNFOEC02] consider possible architectures for ring to mesh migration, and conclude that rings, point-to-point structures, and mesh, would co-exist in the future network. Equipment vendors have [JoMu03][Abowd03] suggested the mesh-like interconnection of ring subnets. There are other approaches that have considered a hybrid ring-mesh network [Abowd03]. In the "ring mining" approach, the constituent spans of rings are logically reorganized to support operation of a span-restorable mesh [ClGrDRCN01]. Results showed that this approach could *unlock* enough usable capacity to serve nearly 300% more demand in some test networks. The philosophy or main idea in ring mining is that from a high level view, existing ring networks are still only collections of working and spare fibers and high-speed interfaces or wavelength channels. Usually metro ring networks are only logical entities built upon a largely mesh-like physical network graph. Ring mining seeks to reuse or "*mine*" these existing high speed line systems to form a span-restorable mesh that uses no new capacity, but can support continuing growth through the creation of new networking efficiencies.

5.1.3. The Straddling Span Interface Unit (SSIU)

In this section we explain how a ring ADM can be converted for reuse as a *p*-cycle nodal element. The initial observation is that a BLSR ring is somewhat like a *p*-cycle that has no straddling spans. Conversely, we can view it as an incomplete part of a whole *p*-cycle, i.e., just the substructure that provides the circumferential protection cycle and the mechanism to cope with "on-cycle" failures. One possible mechanism to convert an ADM to a *p*-cycle node uses the "extra traffic" feature that most BLSR ring systems have. "Extra traffic" is normally a feature that allows the network operator to transport any other lower-priority traffic (in a compatible format for the ring's line-rate signal) over the ring's protection channel. Extra traffic will be bumped off if the ring switches to protect its own working channels. Thus the missing functionality related to straddling spans can be added, effectively allowing reuse of the ring ADM as part of a *p*-cycle node. Figure 5-1 shows a generic ADM or OADM as part of a ring configuration to which a new device is coupled that supports *p*-cycle straddling span access to the ring protection capacity. The only point of physical interface between the new device, called a Straddling Span Interface Unit (SSIU) and the existing ring is where the new device is attached to the "extra-traffic" ports of the ring [GrClLe02]. When the SSIU is attached to the ring's extra traffic ports at the co-located ADM, the normal (non failure) protection channel continuity is then provided by the SSIU, through itself. The existing ring does not strictly need to even know that the straddling span unit is anything other than an apparent source/sink of some form of low priority traffic at its site. More pragmatically, however, an exchange of state information would be required so the SSIU knows when the protection ring is free in each direction, and for the ADMs to be put in protection lockout mode if the SSIU has accessed protection for a straddling failure. In the case of a SONET ring for example, where the full protection state and protection protocol is accessible in the line overhead bytes (K1,K2 in SONET and possible DCC channel management signaling), co-operation of the SSIU with the ring could be entirely transparent to the existing ADM and made possible through the fact that the SSIU needs only passive access to the signaling protocol on the ring protection channel. Alternatively it could be given authority to source/sink protection protocol sequences as needed. Also, because the protection path continuity is through the SSIU, not the ADM, the SSIU can completely observe the status of the protection channel, observe ring switches, and effectively block out or deny ring switches, without affecting them, when needed due to a prior SSIU switch. More specifically, the functions of the *p*-cycle SSIU device as defined in [StGrUS02] and illustrated in Figure 5-1 are:

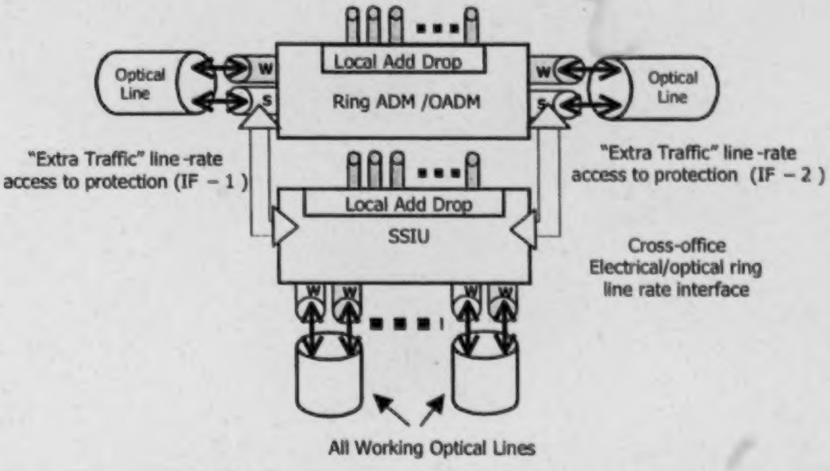

Optical Line
W
S
Local Add Drop
Ring ADM /OADM
W
S
Optical Line
"Extra Traffic" line-rate access to protection (IF – 1)
"Extra Traffic" line-rate access to protection (IF – 2)
Local Add Drop
SSIU
Cross-office Electrical/optical ring line rate interface
W
W
W
W
All Working Optical Lines

Figure 5-1. *p*-Cycle Straddling Span Interface Unit. (Adapted from [ClGrDRCN01])

- To normally connect Extra Traffic IF-1 through to Extra Traffic IF-2 so that the protection continuity of the ring is normally maintained.
- To sense either idle pattern or traffic pattern on protection and/or passively monitor the existing ring signaling protocol so it knows the ring protection status. (In actual products this may also connect the SSIU to the ring-wide internal supervisory LAN, enabling almost any further exchange of control

and status information and development of any new software upgrades to support SSIU-ring interaction).

- Upon failure in the pre-existing ring (an "on-cycle" failure for the *p*-cycle), the *p*-cycle SSIU does nothing except maintain the continuity of the protection channel path through itself. It does, however, note the "in-use" status of the protection channel (as in ii).
- Upon failure of a straddling span, the SSIU interrupts the through-continuity of the protection path of the prior ring and performs BLSR-like loop-back switching to substitute the failed working (bidirectional) signals into the ring protection channels. The SSIU uses the ring protection in both directions to protect pairs of line-rate working interfaces.

An alternative to use of SSIUs is a straight migration to a cross-connect based operation, or custom *p*-cycle ADM-like nodal equipment [StGrUS02], but it seems attractive to reuse the existing ADMs if possible, as SSIUs can also function as an ADM with a specialized firmware upgrade. Note also that SSIUs are added only where the resulting *p*-cycles support straddling spans. All other *p*-cycle nodes require only their existing ADMs.

5.1.4. Implementing *p*-Cycles on a SONET Ring Network using SSIUs

Let us now work through an extended example of how graceful and effective the evolution from rings to *p*-cycles could be in many cases. For a manageable example we consider just two rings in a "matched node"-coupled arrangement from an assumed legacy ring-based network and we note that a "span-elimination" [LeGrMo99] was involved at the time these rings were designed. This is the initial situation shown in Figure 5-2.

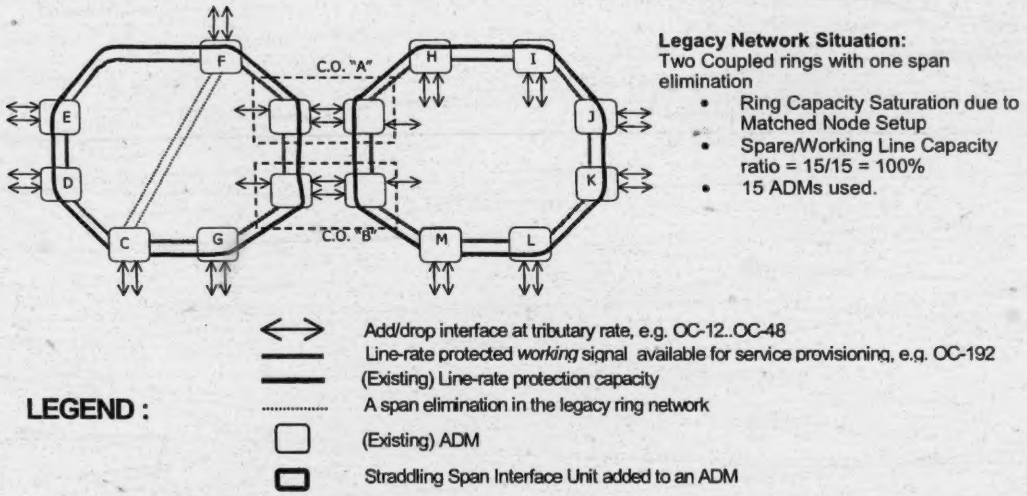
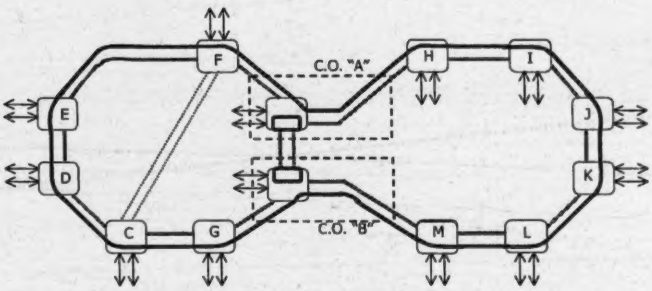

F
C.O. "A"
H
I
E
J
D
K
C
G
C.O. "B"
M
L
Legacy Network Situation:
Two Coupled rings with one span elimination
Ring Capacity Saturation due to Matched Node Setup
Spare/Working Line Capacity ratio = 15/15 = 100%
15 ADMs used.
LEGEND :
Add/drop interface at tributary rate, e.g. OC-12..OC-48
Line-rate protected working signal available for service provisioning, e.g. OC-192
(Existing) Line-rate protection capacity
A span elimination in the legacy ring network
(Existing) ADM
Straddling Span Interface Unit added to an ADM

Figure 5-2. Ring to *p*-Cycle conversion - Legacy Network. (Adapted from [Grov03])

What might trigger the first planning action is imminent exhaust on the "facing" spans between Central office buildings A and B due to "drop-and-continue" capacity consumption. The initial pair of rings employs 15 line-rate optical interface pairs for working and for protection for an initial protection to working capacity ratio of 100%. The first step in evolution is to form a *p*-cycle out of the two rings. We do this by eliminating two of the ADMs where the prior rings interfaced and adding SSIUs to the remaining two ADMs as shown in Figure 5-3.

The *p*-cycle uses the existing ADM line rate interfaces, which used to serve the drop and continue spans, to complete the outer perimeter of the *p*-cycle. The SSIUs support one straddling span of two working line-rate channels where the drop and continue spans used to be. Loss of the new *p*-cycle node will cause outage for demand flow through the new straddling spans. However, demands previously transiting between the legacy rings via drop and continue arrangements have equivalent survivability against node loss because they are routed on the (ring-like) perimeter of the *p*-cycle. No new protection capacity is added and the spare to protection ratio drops to 80%.

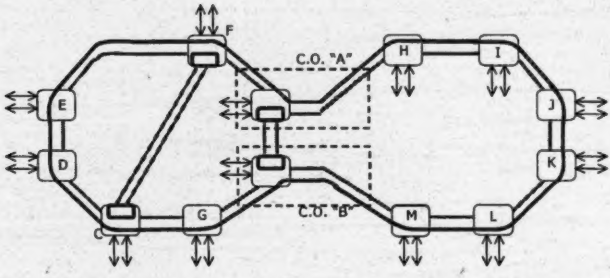


Growth Step 1:
Convert to one *p*-Cycle:
Planning Action:

- Salvage 2 ADMs
- Add SSIUs to two ADMs (Corresponding Node survivability inherent without Drop & Continue penalty anywhere)
- Spare/Working Capacity = 13/15 ~87%
- Exhaust relieved by elimination of Drop & Continue

Figure 5-3. Ring to *p*-Cycle conversion - First Planned Growth Stage. (Adapted from [Grov03])

Next suppose that continued growth of demand routed through the vicinity of nodes D C G B threatens to exhaust the working capacity on one or more spans in that region. This could serve as the trigger for the next planning action as shown in Figure 5-4 — to reinstate the old eliminated span and commission two line-rate working channels on it. This relieves capacity on all the spans mentioned as well as shortening many working routes in the vicinity. The redundancy is now down to ~71%. Again, growth has been served with no additional protection capacity. Investment in SSIUs and added working capacity is made, but this is purely to serve the growth in demand. The legacy protection capacity is being financially leveraged because the prior existing ring protection channels, used as a *p*-cycle, are being stretched to serve more efficiently.

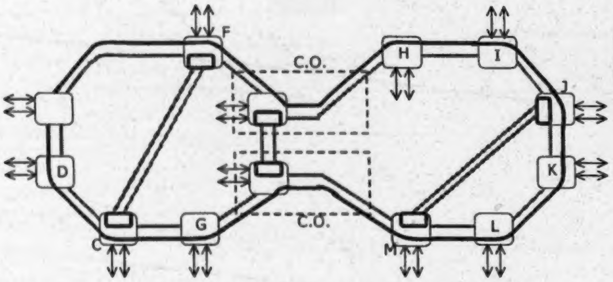


Growth Step 2:
Exhaust from growth forecast in regionof E D C G
Planning Action:

- Re-instate the "eliminated span"
- Spare/Working Capacity = 13/17 ~76%
- Add two more SSIUs
- Returns several working routes to shortest paths relieving working capacity in left hand side

Figure 5-4. Ring to *p*-Cycle conversion - Second Planned Growth Stage. (Adapted from [Grov03])

In the final step in Figure 5-5 we postulate similar accumulating growth in the right hand region of the example and show how the operator might again respond



Growth Step Exhaust pending from growth region of M L

Planning

- Acquire or lease additional capacity -M
- Spare/Working Capacity 13/19 ~68%
- Add two more
- Returns several working routes shortest paths relieving working in right hand

Figure 5-5. Ring to *p*-Cycle conversion - Third Planned Growth Stage. (Adapted from [Grov03])

efficiently – in this case by leasing or otherwise establishing two new working-only lightpaths between nodes M and J, or more generally, acquiring their own new M-J span. At this stage we have grown the network into a mesh-like redundancy of 63.2% and only spent money in this evolution on equipment and capacity directly needed to serve demand growth, and only when and where it actually materializes.

5.2. Ring-Mining Design Models

5.2.1. Ring Mining Without Capacity Addition

The simplest first step is to find the largest common multiplier (λ) that can be applied to every element of the demand matrix[xv], while still keeping the network restorable using *p*-cycles, without adding any capacity at all. This is addressed with the Mixed Integer Program (MIP) below which is a variation of those methods in [ClGrDRCN01] and the Joint Capacity Assignment (JCA) model for *p*-cycles from [GrDoLEOS02]. The formulation jointly optimizes working and spare capacity for a *p*-cycle based network under the existing ring fiber capacity limits – also called pure ring mining.

Sets:

S Set of spans between mesh cross connection points

Q^r Set of eligible working routes available for working paths on end-to-end demand relation r

D Set of all point-to-point (*active*) demand quantities, indexed by r

X	Set of eligible cycles on the network graph, that can form *p*-cycles
M	Set of module-types that are available

Parameters:

d^r	Number of demand units between end-node pair *r*,
π_j^x	1 if cycle *x* lies on span *j*, 0 otherwise
$\zeta_j^{r,q}$	Takes the value of 1 if the q^{th} working route for the demand pair *r* goes through span *j*
Z^m	Capacity of m^{th} module type
C_j^m	Cost of using module type *m* on span *j*
t_j^m	Number of modules of type *m* available on span *j* from earlier ring design
ρ_i^x	Number of restoration paths provided by an instance of cycle *x* for restoration of span *i* (0, 1, or 2)

Variables:

$g^{r,q}$	The number of working capacity units required on the q^{th} working route to satisfy the demand between node pair *r*
n^x	Number of unit capacity copies of *p*-cycle *x* required in the final design
s_j	Number of units of spare capacity allocated on span *j*
w_j	Number of units of working capacity on span *j*

Objective Function:

$$\text{Maximize } (\lambda) \qquad (7)$$

(Maximize the minimum demand growth possible for all demands)

Constraints:

$$\sum_{q\in Q^r} g^{r,q} = \lambda \cdot d^r \qquad \forall r \in D \ \lambda \geq 1 \qquad (8)$$

(All demands must be routed)

$$\sum_{r\in D}\sum_{q\in Q^r} \zeta_j^{r,q} \cdot g^{r,q} \leq w_j \qquad \forall j \in S \qquad (9)$$

(Place enough working capacity to support the demand)

$$\sum_{x\in X} \rho_i^x \cdot n^x \geq w_i \qquad \forall i \in S \qquad (10)$$

(Place enough *p*-cycles to protect all working capacity)

$$\sum_{x\in X} \pi_j^x \cdot n^x \leq s_j \qquad \forall j \in S \qquad (11)$$

(Place enough spare capacity to build all *p*-cycles)

$$w_j + s_j \leq \sum_{m\in M} t_j^m \cdot Z^m \qquad \forall j \in S \qquad (12)$$

(Only- allowed to re-use existing ring capacity)

Constraints (8),(9) ensure that all working demand is routable even after applying a uniform growth multiplier λ to each demand. Constraints (10),(11) ensure that all the working capacity on each span is restorable and that there is sufficient spare capacity for the chosen *p*-cycles to be formed. Constraint (12) ensures that the formulation only uses the existing capacity on each span – mined from the prior ring networks. The objective function (7) finds the maximum uniform demand growth multiplier possible within the capacity of the existing ring spans, while simultaneously ensuring that the network is routable and restorable.

5.2.2. Ring Mining With Strategic Capacity Addition

We now allow modular capacity to be selectively added while mining rings to *p*-cycles. The demand growth multiplier λ now becomes a parameter to the problem and is increased in fixed steps, to calculate the minimal extra capacity addition required to enable each step value of λ. The new MIP formulation optimizes for minimum cost of adding modular [DoGrJSAC00] capacity. Module costs consider a 4x2x economy of scale. The objective function is therefore:

$$\text{Minimize:} \quad \sum_{j\in S}\sum_{m\in M} n_j^m C_j^m \qquad (13)$$

(Minimize the total number of modules of capacity added to the network)

We consider all the constraints (8)(9)(10)(11) with constraint (12) modified to (14) below.

$$w_j + s_j \le \sum_{m\in M} (t_j^m + n_j^m)\cdot Z^m \qquad \forall j \in S \qquad (14)$$

(Allowed to strategically add modules of capacity to the network)

The new objective function minimizes the cost of adding capacity. n_j^m is the number of modules of type *m* strategically added to span *j*. Constraint (14) limits the working + spare capacity total on each span to be under the capacity available from the existing ring network, plus the added modular span capacity. Adding capacity increases the value of the objective function, and thus ensures that capacity addition is considered only after fully exploiting the existing ring capacity. Minimizing the cost also ensures that the solver chooses the least costly construction of modular capacity. For example, adding 1 OC-48 module would be less expensive than 4 OC-12 modules on the same span.

5.2.3. Ring Mining Without Re-Arranging Existing Traffic

In the models so far, working routes are jointly optimized with the placement of *p*-cycles. This is fine for growth demands and assumes that any implied rearrangements of existing paths may be acceptable if coordinated with customers. But, if we wanted to consider a strict prohibition against rerouting any existing demand, we can put it into the planning model from 5.2.1by changing Eq. (8) to Eq. (15) and Eq. (14) to Eq. (16), while the objective function is the same.

$$\sum_{q\in Q^r} g^{r,q} = (\lambda - 1)\cdot d^r \qquad \forall r \in \mathbf{D} \;\; \lambda \ge 1 \qquad (15)$$

(New demand growth is jointly routed)

$$w_j + s_j + w_j^0 \le \sum_{m \in M} t_j^m \cdot Z^m \qquad \forall j \in S \qquad (16)$$

(Demand growth can be routed on added capacity)

where w_j^0 is the previously existing working capacity on span j. All other constraints (8)(9)(10)(11) also apply. Constraint (15) ensures that only the new growth demands are considered for routing decisions that are jointly coordinated with the spare capacity placement decisions. The existing working capacity on each span is left untouched because of constraint (16), thus preventing any routing change for existing demands.

5.2.4. Ring-Mining With Strategic Capacity Addition and No Rearrangement of Existing Traffic

When selective capacity addition is considered along with the constraint on existing working routing, λ becomes a parameter to the problem, increased in steps and the problem type changes to a minimum cost design as earlier in section 5.2.2.

Minimize: $$\sum_{j \in S} \sum_{m \in M} n_j^m C_j^m \qquad (17)$$

(Minimize total capacity addition)

With constraints (15), (8)(9)(10)(11) and (16) being changed to (18) below.

$$w_j + s_j + w_j^0 \le \sum_{m \in M} (t_j^m + n_j^m) \cdot Z^m \qquad \forall j \in S \qquad (18)$$

(Add modules of capacity without affecting existing capacity.)

Constraint (18) ensures that the existing working capacity is left untouched. The existing spare capacity is re-optimized for *p*-cycles, including the option to add new capacity.

5.3. Experimental Setup

5.3.1. Initial Ring Network Designs

For quantitative studies on ring-mining, it is essential that we have some representative ring designs of typical multi-ring networks as input to the problems. We used two sets of data for the results in this Chapter.

5.3.1.1 Pseudo Random Test Networks (Morley Networks)

The first set of 17 networks are from a recent Ph.D. thesis [Morley01] on ring design methods. These research based designs are well-loaded and balanced. Real-world ring networks are not as well-designed as these. All designs serve the full demand matrix for which they were designed and the same demand matrices are provided as input to the ring-mining formulations. Six of the ring designs were produced using a fixed charge and routing IP (FCRIP) ILP model from [Morley01], where the rings and the working routing is chosen jointly. A Span Coverage IP (SCIP) method was used to first route the demands along the shortest path and then a minimum cost ring cover is found. The last five test networks use a Tabu Search heuristic called RingBuilder [Morley01]. Span eliminations [LeGrMo99] are only exploited in the FCRIP formulation. There are three different input graphs shown in Figure 5-6 for each ring network design and each serves the complete demand matrix. More data on these networks is available in [ClGrDRCN01].

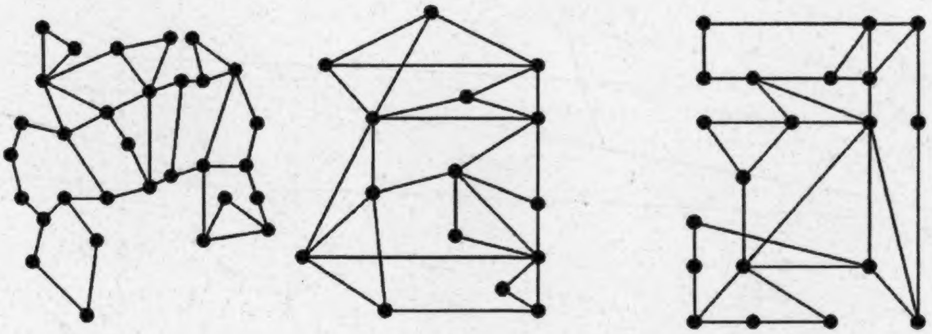

(a) Net 32 (b) Net 15 (c) Net 20

Figure 5-6. Three network graphs used to produce the input ring designs. (Adapted from [Morley01])

5.3.1.2 TELUS Case Study

As part of the overall project from which this Chapter arises, we collaborated with TELUS on a larger study of migration of real-world legacy networks of ring and point-to-point links, to *p*-cycles. The case study network is the TELUS Calgary Metro Optical Network that has an assortment of 46 transport systems consisting of 1 OC-48 point-to-point link, 6 OC-48 BLSR rings, 5 OC-48 UPSR rings, 12 OC-12 BLSR rings, and 22 OC-12 point-to-point links. The physical network has 31 nodes and 48 spans, while the logical network view has 31 nodes and 87 links. A total traffic volume of 683 STS-1 service paths is served by a capacity investment equivalent to 6792 single hop STS-1 channels. A characteristic of this metro network, shared by many metro networks is the high degree of demand capture of each of these rings. Approximately 85-90% of demands do not transit from ring to ring. This happens because a metro network is planned and built incrementally in planning cycles. Each planning cycle adds rings to the metro optimally. This results in the overall network being sub-optimally utilized, as free capacity does not necessarily exist in a continuous ring-like fashion. The metro network then becomes simpler to manage, and involves less ring-transit costs, but has a significant amount of stranded capacity. An isomorphic graph of the Telus Metro network is in Figure 5-7. Figure 5-7(a) shows the actual physical facility routes that contain stub nodes. Demands originating or terminating at these stub nodes and the nodes themselves are removed from the graph for survivable routing as it is physically impossible for a demand terminating at a stub node to survive a single cable cut on the span adjacent to the stub node. The resultant logical diagram of all current demands is shown in Figure 5-7(b). Figure 5-7(b) shows a rich demand matrix indicating the heavy load on the metro network. Such dense connections are typical of metro networks.

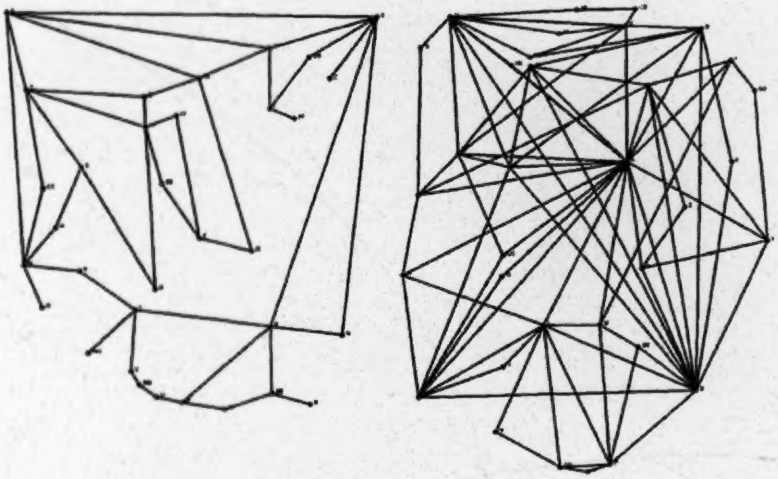


Figure 5-7. Isomorphic Graph of the Telus Calgary Metro Network (a) Actual physical facility and (b) Logical connections

5.4. Results and Discussion

5.4.1. Ring Mining Without Capacity Addition

The results obtained for the 17 test networks with the formulation in 5.2.1 are compared to results obtained previously for mesh target networks (for the same test networks) in [ClGrDRCN01]. Both the formulations - rings to mesh and rings to *p*-cycles were solved to within 5 percent of the optimal solutions. The rings to span restorable mesh formulation was offered a choice of routes with hop length limit of 5, while 1000 candidate *p*-cycles were chosen by the pre-selection methods in [GrDoLEOS02] for the formulation here. Table 3 shows the maximum feasible multiplier values and their comparison to growth factors for mesh networks. These results establish that *p*-Cycles are able to sustain growth multipliers similar to the corresponding values obtained for mesh. A seemingly odd discrepancy is observed in the results in Table 3 for cases 5, 11, 13 and 17. These cases show that migrating to *p*-cycles would permit higher growth multipliers than migrating to a span restorable mesh. From a theoretical point of view, this is not possible as the complete solution space for *p*-cycles is always a subset of the complete solution space for mesh. However, to keep the problem solvable in reasonable time, it becomes necessary to pare down the input set of eligible restoration routes or candidate cycle set.

Table 3. Max Uniform Demand Growth Multiplier. (Adapted from [KoGrNFOEC03])

Network	Mesh	*p*-Cycle	Network	Mesh	*p*-Cycle
1	1.24	1	10	2.24	2
2	2.91	2.75	11	1.07	1.29
3	1.05	1	12	2.05	2
4	1.16	1	13	1.36	1.54
5	1.02	1.3	14	2.91	2.75
6	2.05	2	15	1.31	1
7	1.32	1	16	1.24	1.2
8	2.91	2.75	17	1.38	2
9	1.13	1			

This creates some inherent incomparability when the number of eligible cycles to the *p*-cycle formulation is limited by the pre-selection heuristic [GrDoLEOS02] while the mesh restoration formulation has a hop limit restriction of 5 on eligible restoration paths. For example a 7-node *p*-cycle can restore an on-cycle span failure using more than 5 hops, and can thus consider restoration paths that are not allowed to the mesh restoration formulation. This is not an anomaly or error, but is simply caused by the closeness of the *p*-cycle and span restorable mesh solution, and because of the differing metrics used. This motivated the entire study in Chapter 6 that helps understand these anomalies.

For the TELUS case study network in Figure 5-7, every demand in the metro network can be scaled up 1.5 times before the need to add more capacity arises. Additionally, the *p*-cycle solution for the 1.5 scaling of the demand matrix assigns 85.4% of the available capacity. Upon visually inspecting the physical graph, it is observed that there are a good number of degree 2 nodes, typical for a ring-based metro network. Capacity utilization may therefore be less than 100% because of intermediate span exhaust. For example, if all the capacity on the possible working routes between an origin and a destination is already assigned (which can be found from a min-cut of the capacitated network graph between the origin and the destination), a blocking condition results. No matter how much capacity exists on the other spans of the network, capacity addition would be required to route any growth of demand through exhausted spans. This leaves some of the capacity unusable, and stresses the value of allowing selective strategic capacity addition to fully exploiting all other existing capacity.

Primarily, however, these results establish that the capacity efficiency with *p*-cycles is almost as promising in general and quite comparable in some networks to that of span-restorable mesh networks as the ring mining target. This "pure ring mining" potential is, however, just an indicator. In practice we can do better if some capacity additions are allowed during migration. Thus we take the next step by allowing selective capacity additions to the network and considering migration to *p*-cycles.

5.4.2. Ring Mining With Strategic Capacity Addition

Sample results for two of the test networks are in Figure 5-8. For simplicity Table 4 shows the total number of added "systems" needed to reach the corresponding growth factor, a system being either an OC-12 or OC-48 modular capacity addition. Network 6 only requires 30 modular system additions to grow more than 300% but no additions at all until a doubling in demand served is already sustained. This capacity addition profile is very close to the profile obtained for span restorable mesh in [ClGrDRCN01]. Other networks also exhibit similar growth patterns. The corresponding solution for the TELUS network is in Table 4.

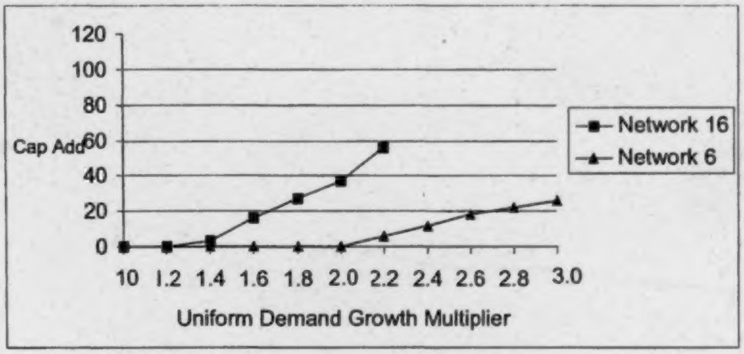


Figure 5-8. Capacity Addition Profile for two test networks growing under ring mining. (Adapted from [KoGrNFOEC03])

Table 4. Capacity addition required with change allowed to working routing. (Adapted from [KoGrNFOEC03])

λ	Additional Modules Required		% Capacity Added
	OC-12	OC-48	
1.0	0	0	0%
1.5	0	0	0%
1.6	1	0	0.20%
1.8	2	1	1.24%
2.0	4	4	4.1%
3.0	1	33	27.5%

In Table 4, we can see that for a relatively minor investment (adding 4 OC-12 and 4 OC-48 modules on selected existing spans), the network operator can *double* the carrying capacity of this network. Existing capacity utilization now increases to 87.8%. Even though there is a popular school of thought that considers capacity essentially free, re-using existing capacity more efficiently still makes better business sense. Even if new dark fiber is free, there is still a real cost-inefficiency associated with adding new line cards, ADMs, lasers, etc., if not otherwise needed. Serving 3 times the demand requires 33 OC-48 modules to be added, but even though 33 OC-48 modules are added, going only by capacity values, still only about 91% of the previously existing fiber capacity is re-assigned. This happens because the network runs out of un-assigned capacity on critical spans when a very high demand multiplier is considered. Such high demand growth multipliers literally force the addition of new capacity on every span in the working path, while still rendering some capacity too expensive to be used with the present demand matrix. This seeming waste of capacity is because a straight out uniform demand multiplier, though useful to establish the potential benefit of ring mining, may not mirror the actual demand forecast planned for this network. An accurate demand forecast, incorporated into a ring mining study should give better results. If selected demands are allowed to grow while others are kept constant, and new O-D pair demands are simultaneously introduced, a corresponding minimum-cost evolution plan for a specific network can be worked out using the minimum capacity addition model.

If instead of requiring a uniform demand growth multiplier over all demands, we allow all individual demands to grow to an individual maximum λ within the ring capacity constraints, and with minimum cost, the optimal solver almost always assigns maximum growth to physically adjacent node demand pairs, and then to next shortest demand pairs and so on, which makes sense given how the model is formulated, but may not correspond to the actual future growth pattern of the network.

Finally, there is no general pattern as to why a particular network requires a specific number of modules added for demand growth but the important effect is the deferral of cost until significant growth occurs. So far we have established that *p*-cycles and span restorable mesh are comparable in capacity efficiency for migrating from ring networks. What is not quite evident in these numeric results is the potential architectural benefit of using *p*-cycles. *p*-Cycles would have similar capacity requirements as mesh, but can be implemented on ADM–SSIU based networks. Thus, using *p*-cycles would give the added advantage of being able to defer DCS equipment purchases.

5.4.3. Ring Mining With Strategic Capacity Addition Without Affecting Current Working Traffic

Even with this added constraint in the TELUS network we observe that the metro network can support a uniform demand growth multiplier of **1.33** without any capacity addition and without rearranging the existing working demands. Network capacity utilization is at 84.3%, which is not very much less than the *p*-cycle solution that allowed working routing changes. This shows that even if the carrier chooses to migrate without affecting existing customers there are still considerable efficiencies to be gained.

5.4.4. Ring-Mining With Strategic Capacity Addition With Possible Rearrangement of Working Traffic

The results are shown in Table 5. As expected, more capacity is needed because of the imposition of the fixed working routing condition. Now, instead of 8 (from Table 4), the service provider has to add 10 system modules, on existing spans, to achieve a doubling of carrying capacity. Considering the ratio of capacity costs to nodal equipment costs, this is again not that unreasonable an additional capacity requirement, if we consider the potential benefits of migrating to *p*-cycles without affecting existing customers and without buying any DCSs. At the same time, the new demands can be served far more efficiently than with the prior ring networks. Additionally, adding these extra systems module allows 89.35% existing capacity utilization, which is higher than the previous value of 85.4%.

Table 5. Capacity Addition required with no change allowed to working routing. (Adapted from [KoGrNFOEC03])

λ	Additional Modules Required		% Capacity Added
	OC-12	OC-48	
1.0	0	0	0%
1.33	0	0	0%
1.5	1	1	1.53%
1.6	3	1	2.15%
1.8	2	4	5.53%
2.0	4	6	8.60%
3.0	8	38	49.1%

5.5. Summary

We have established that *p*-cycles are almost as capacity-efficient as span restorable mesh for ring mining. The nodal compatibility of *p*-cycles and ring networks would be the prime mover behind *p*-cycle implementation in the metro environment. Network operators can go on using their existing network channel capacity for a significant period of time without adding any new capacity or replacing nodal elements with DCSs. *p*-Cycles offer the option of being able to serve increasing traffic demand, with constraints on re-routing existing working demand. This cost-effective migration strategy, which allows the service provider to migrate to *p*-cycles without affecting existing customers, while simultaneously deferring the costs of serving new growth, could help ensure the survival of the network operator through uncertain and competitive economic times.

Chapter 6. PATH LENGTH CONSTRAINTS IN *P*-CYCLE NETWORK DESIGN

In this Chapter, we present a novel approach for imposing direct restrictions on the length of *p*-cycle protection paths as opposed to overall size restrictions. This allows the optimal solver to consider path length restrictions in a very direct and specific way. This new approach allows us to accurately compare path or hop-limited *p*-cycle and mesh designs. This new method also allows us to investigate whether *p*-cycles also exhibit a threshold hop limit effect like mesh networks.

6.1. Introduction

Fundamental studies on network architecture often consider designs that are restricted only by the network graph itself and the requirement of restoration paths being loop-free. It has, however, always been understood that a maximum hop or maximum distance limit may often pertain in practice. In a WDM optical network the main interest in asserting such limits is to ensure transmission integrity for paths that are re-routed for protection. Such hop limits are not very naturally or easily enforced in the pure arc-flow (nodal trans-shipment) formulations for spare capacity placement for mesh restorable designs (see [Grov03] (pp. 294-296) for examples of such models), but in the path-flow assignment approach by Herzberg [HeByTON95] and many of its extensions in [Grov03], it is easy to include such limits in the design. This is done simply by restricting the set of eligible restoration routes for each failure scenario to those within the desired hop, distance or any other limiting criteria [HeByTON95],[IrMaTON98], [RaMuINFOCOMM]. Early protocols for distributed "self-healing" restoration [Grov97] and for distributed self-planned protection [Grov94] also both inherently permit user-specifiable limits on replacement path lengths. This is also an obvious capability of GMPLS-based constrained routing protocols when used to find a replacement path, or to activate a pre-planned backup path.

Herzberg also demonstrated the phenomenon of a "threshold hop limit" in the design of span-restorable (or equivalently span-protected) mesh networks. As the hop limit used in the design process is increased, it reaches a threshold value above which no further reduction in total spare capacity requirement is reached. At and above the threshold hop limit, the opportunities for spare capacity sharing over disjoint failure scenarios are in effect saturated on the given graph topology. For example, in the relatively well-connected and uniform "Bellcore" network in [ScGrICC02] this occurs at a hop limit of five, at which point the network redundancy is a rather attractively low 51%. This means that no restoration path would ever have to be longer than five hops in this network, and that if up to five hops is permitted, optimal capacity efficiency is also achieved.

Not all networks exhibit such a low threshold hop limit, however. In work on sparser topologies we have seen threshold hop limits of up to 12, although this is almost always a result of many degree-two nodes in chain-like subnetworks. (In the meta-mesh abstraction [GrDoJSAC02] of those networks, the effective threshold hop limit can more often be expected to again be in the five to seven hop range, say.) Advantages to restricting the hop limit as much as possible in survivable design are numerous and obvious and include increased availability (fewer network elements involved in restoration), reduced regenerator cost associated with protection paths, and depending on signalling protocols, possibly increased restoration speed from fewer nodes having to react, and decreased regeneration costs. Especially for an optical network, all other things being equal (such as the total spare capacity needed) shorter protection paths are always preferred. Some other work on path length restriction tactics in optical mesh-restorable networks is in [DoGrOFC01].

A natural, if somewhat theoretical, interest motivated by past work on span-restorable mesh networks is therefore to ask "what about hop-limit considerations for p-cycles?" From several standpoints, *p*-cycles are the close cousins of the span-restorable mesh architecture. In the pre-planned mode, both activate protection paths that extend between the immediate end nodes of the failure only, and p-cycle designs are empirically closest in efficiency to span-restorable networks. For this reason it would be useful to understand if *p*-cycle networks also exhibit a threshold hop limit effect. And if so, is it characterized in terms of a threshold limit of cycle circumference, or in some other way? These are the two research questions that are studied in this Chapter.

This Chapter is based on collaborative work with Anthony Sack, currently with Consumer Broadband Division, TELUS Communication. Much of the material herein is based on previous publications at IEEE-BROADNETS 2004 [KoSaBROADNETS04] with an invited Journal publication in [KoSaOSN]. The conference version of this paper in [KoSaBROADNETS04] won the IEEE-Phillips Best Paper Award at BROADNETS 2004.

6.1.1. Overview of Path Length Restrictions

Optical networks are always designed with a maximum un-regenerated optical path length restriction. As discussed in Chapter 4, this limit basically defines how long the signal can travel before it accumulates an unacceptable amount of noise. This reach, as we discussed in Chapter 4, depends on many factors such as the type of fiber, the power launched into the fiber by the laser, the SNR of the laser, the minimum SNR requirement of the photo detector, the number of active wavelengths on the fiber, etc. In addition, regenerators are essentially full scale OEO cross-connects and therefore are very expensive. Network operators tend to operate an optical network such that all paths (whether working or protection) are under the optical reach length limit. Shorter paths also improve the general availability of the working path. Regardless of what the reasons really are, it seems like a reasonable expectation for *p*-cycle design tools to have controls on the maximum protection path lengths. Such restrictions are easily built into design methods for span restorable mesh network design as a fairly straightforward restriction on the maximum eligible route length. Extensions to path restoration also incorporate this length constraint [IrMaTON98]. In both cases, the pre-processor simply discards the routes that do meet the specific criteria – such as max hop limit or max length limit. Doucette et al. in [DoGrOFC01] examined length limits for mesh networks, but to date, no corresponding work has been done for *p*-cycle network design.

6.1.2. Circumference vs. Hop Limits For *p*-Cycles

Restricting the total number of cycles provided as input to the problem is a necessary first step to any optimal *p*-cycle design. This ensures that the ILP solver has a small enough solution space and can obtain solutions within a reasonable time. The simplest cycle pruning heuristic is to only choose the shortest 1000 or 2000 cycles from the set of all cycles. This is the approach used in initial works on *p*-cycle design such as [GrStICC98]. In more recent work, newer cycle selection metrics such as the TS and AE (as discussed in [GrDoLEOS02]) have been used as cycle selection metrics. Yet other work by Schupke et al. [StGrICC04] used the dual failure susceptibility of a cycle as a selection metric for cycles. But as a class, all these cycle size limiting techniques had the primary goal of improving the efficiency and reducing the run-time of the solution. In [StGrICC02], for the first time, the authors studied the effect of cycle size limiting on the capacity efficiency of *p*-cycles and, as expected, as the maximum cycle size is lowered, the capacity efficiency of the solution worsens. Still, the solutions in [StGrICC02] exhibit an attractively low redundancy of 51% even at very low hop limit restrictions.

To date no attempts have been made in the *p*-cycle literature to specifically limit the path length in *p*-cycle networks, other than to use an indirect cycle size limiting. Limiting the maximum cycle size is an effective means to guarantee that no protection path can be longer than the size of the biggest cycle admitted to the input to the solver. In ring networks, restricting the ring size is the same as restricting protection path lengths. In *p*-cycle networks, straddling spans are protected segments on the cycle. A large cycle could, quite effectively, provide short protection paths for a straddling span. This protection path would in effect be acceptable from an optical path engineering point of view, even if the cycle in itself is too long. The basic idea is that one may admit a large cycle into the problem but prevent it from protecting any on-cycle spans. Of course, the solver would be biased toward selecting cycles that protect both straddling and on-cycle spans, but in the specific case where a large cycle is justified for some suitable set of straddling spans, it must be allowed to be part of the input cycle set. Circumference limiting therefore is just a method to upper bound the maximum protection path length, with the side effect that short protection paths provided by large cycles to straddling spans are excluded from the problem.

This is a real problem when comparing *p*-cycles to span restorable mesh networks. In flow-based mesh design ILPs, constraining the maximum path length is a simple problem as one only needs to prune the input eligible route set so that none of the routes are longer than a specific constraint. It is therefore not possible to accurately compare a mesh design where path lengths are explicitly specified to a *p*-cycle design where path lengths are indirectly specified as cycle size limits. Thus the only way to make fair comparisons is to enumerate all possible routes in the mesh design and all possible cycles in the *p*-cycle design. But this usually is not a practical solution as the run-times would sometimes make these problems infeasible to solve.

6.1.3. Motivation and Current Goals

The study of path length constraints in *p*-cycle networks was initially motivated because it gives us the ability to accurately compare *p*-cycle and mesh network designs. This is in itself a much needed capability as many comparisons have been made between the two architectures in the literature. From an empirical standpoint we can create examples where precisely limiting the path length in *p*-cycles is better than overall cycle size limits. But what we do not know, prior to actually designing a network with these constraints built in, is whether this extra precision makes any difference at all. The point being, how many times would the opportunity of using a large protection cycle to provide short protection paths be actually used by the solver? Results either way would be very interesting to see.

6.1.4. Threshold Hop Limit Effects in Mesh Networks (Herzberg-Bye)

Herzberg reported that span restorable mesh networks exhibit a "threshold hop limit" effect. What this means is that when a span restorable mesh ILP solver is provided progressively longer routes as part of the eligible route set, (starting from the length limit where the solution is first feasible), the gains in capacity efficiency do not consistently improve. At the threshold hop limit, (the actual numerical value depends on the network and demand details), there is an abrupt 'flattening' of the curve and no more reductions are to be obtained in the total cost of the network. For example, in the relatively well-connected and uniform "Bellcore" network, this threshold limit is '5', at which point the network redundancy is an attractively low 51%. This means that no restoration path needs to be longer than five hops and if all routes up to and including five hops are enumerated, the solution has achieved near optimal capacity efficiency. Allowing routes longer than the threshold significantly increases the run-time and complexity of the ILP without improving solution quality. What Herzberg established is that as long as all routes that have lengths up to and including the threshold hop limit are enumerated for the ILP, the solution is already within a few percent of the best possible solution. For large and dense networks with hundreds of nodes and spans, this is a very important conclusion as the network designer can considerably cut down the run time without affecting solution quality significantly. It is now of interest to ask, do *p*-cycles also exhibit the same threshold hop limit effect? If yes, how do these hop limits correspond with equivalent mesh solutions ?

6.1.5. Hamiltonian *p*-Cycles

The final motivation for examining path lengths comes from recent debate on Hamiltonian *p*-cycles and path lengths. Hamiltonian cycles are those cycles that pass through every node in the network. In some networks, using a single Hamiltonian cycle may suffice for all survivability needs and depending on the working capacity distribution, such a solution may approach the theoretical minimum capacity redundancy of a span restorable mesh network [SaGrNETWORK04], [HuCoCOMMAG02] and [RyPaICOIN]. Hamiltonian cycles are unique in that one may, by construction, create a Hamiltonian cycle that is highly efficient and matches corresponding mesh network designs. But while Hamiltonians are a capability of the *p*-cycle architecture, there is no limitation that only Hamiltonians be used as *p*-cycles for efficiency. In fact, recent work in [ShYaPNET04] [ShMaISCC04] actually incorrectly describe *p*-cycles as a concept that employs only single large Hamiltonian *p*-cycles for protection. They then go on to claim that since these cycles are |N| hops long, they can not provide protection paths that meet optical path integrity restrictions. In this work, we therefore attempt to correct these misconceptions. It is already accepted and understood that in most real-world networks, it is a set of *p*-cycles of varied sizes that is most efficient. Yet what we will demonstrate is that large cycles do not necessarily provide long protection paths. We also develop the necessary design theory to enable fine-grained controls on path length and cycle size. Properly understood, Hamiltonian cycles are an interesting, if academic, special case that enable networks to reach limiting efficiency simply by construction.

6.1.6. The Potential Efficacy of Path Length Restrictions in *p*-Cycle Based Networks

As a first step, it is useful to show a quick empirical "existence proof" illustration that strict path length restrictions are indeed beneficial to the overall network design. The construction in Figure 6-1 from [KoSaOSN] shows how exact path length constraints can outperform (in terms of spare capacity requirements) the prior cycle circumference limiting method.

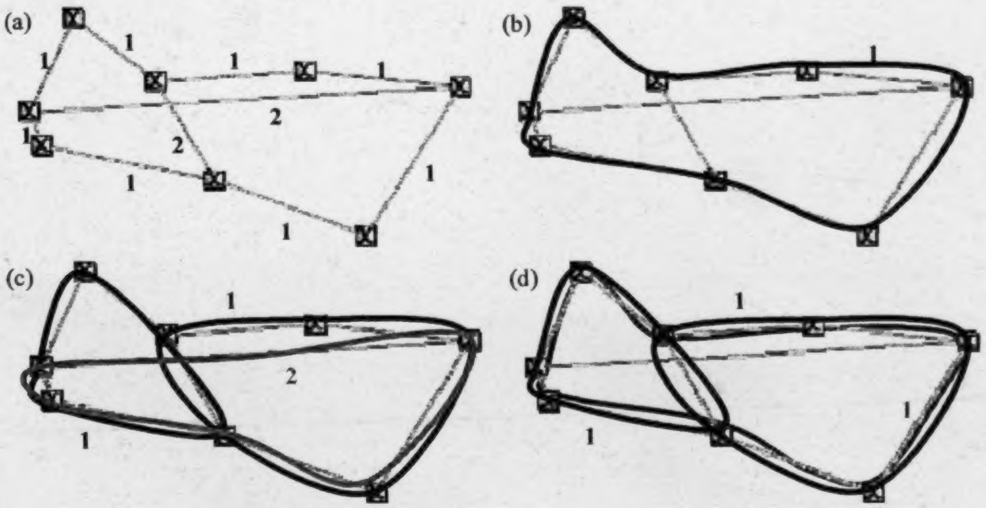


Figure 6-1. Existence proof example of potential benefits of explicit path length restriction versus cycle size limiting. (Adapted from [KoSaBROADNETS04])

Figure 6-1(a) is a capacitated network graph with working capacities indicated along the corresponding spans. The single *p*-cycle in Figure 6-1(b) is the best protection solution for this network using only 8 units of spare capacity to protect 12 units of working capacity (66% capacity efficiency). But the protection paths for on-cycle spans in this particular case are 7 hops long, which for the current example, lets say, are not acceptable. If the acceptable maximum path length is five and we use the circumference limiting method, a possible solution is in Figure 6-1(c) with three small *p*-cycles. The network now needs about 20 units of spare capacity. But if large *p*-cycles are allowed, with the condition that they may not be used to provide any protection paths longer than 5 hops, the solution in Figure 6-1(d) is obtained. One of these cycles is longer than five hops, but it is assigned for protection of only the straddling spans that each bear two units of working capacity. Each straddling span sees a four hop protection path which is acceptable. This particular design has 18 units of spare capacity. While the capacity savings are not huge, this example in Figure 6-1 just illustrates that in theory, use of precise path lengths may be a more efficient solution that just circumference limiting.

6.2. ILP Design Models

The example in Figure 6-1 showed a contrived example of how path length restrictions may improve the solution, to find out whether such savings are possible in practical real-world networks, we use an optimal design model that incorporates direct restrictions on path length. This model is based on the *p*-cycle Modular JCP model published in [GrDoLEOS02] with the AMPL model from [KoGrNFOEC03]. The model makes working path routing decisions jointly with the placement of *p*-cycles. The model is extended so that cycles of any length can be used but no *p*-cycle can protect a span if the protection path is longer than the prescribed hop limit. In other words, regardless of the size of the cycle, protection paths can only be used in a way that respects the hop limit in effect.

We begin by explaining how these path length constraints are added to the previously published *p*-cycle MJCP models [GrDoLEOS02].

6.2.1. Adding Path Length Constraints

Path length constraints are added by generating parameters that specify the lengths of the protection path options offered by each candidate cycle to every span. This is the main extension in this model. If a failure is on-cycle, then the parameter value is the same as *C*-1. If the failure is on a straddling span, then two associated parameters are used to record the lengths of the two protection paths available to the failed span as Figure 6-2 illustrates.

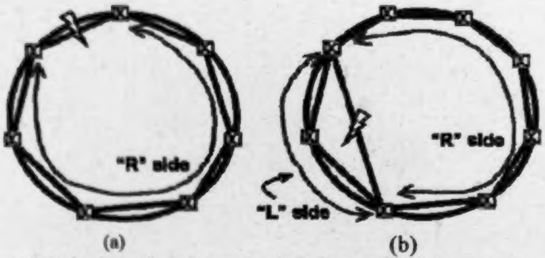


Figure 6-2. Left and right path definitions for (a) on-cycle spans and (b) straddling spans. (Adapted from [KoSaBROADNETS04])

To describe these two new parameters let us first define the notation necessary to denote the different sides of the cycle. Since we want to be able to specify the precise length of each protection path offered by the cycle, instead of just the size of the cycle, we need to consider the two sides separately. This is to ensure that a cycle is potentially used even if only one of its sides offers an acceptable protection path to a straddling span. For straddling span, we define that the left or "L" side of the *p*-cycle is the one that offers the shorter protection path while the right or "R" side is the longer arc of the cycle. For an on-cycle span, the surviving arc of the cycle is always referred to as the "R" side.

As input to the problem we generate a set of eligible cycles of the network graph. But at the same time, despite our extension, we still limit the total number of cycles input to the problem to ensure feasibility and because of the sheer number of possible cycles in a large network. As a purely practical matter, we limit the cycle size to be much higher than the hop limit. So for a test case where we want to limit the hop limit H to five, we may restrict the circumference of the cycle C to about 16 or 18. This is not contradictory to the aim in this study because generally as long as $C>>H$ for the input pre-processing stage, this is not expected to affect the quantitative results.

Next we generate the R and L parameter lengths that specify the protection path lengths that a candidate cycle may offer to a span. Given this nomenclature, the model listing is as follows:

Sets

Z Set of available module capacities, indexed by m.

S Set of spans, indexed by i (failed) or j (surviving).

$\boldsymbol{D}$	Set of demand relations, indexed by r.
$\boldsymbol{P}$	Set of eligible cycles, indexed by p.
$\boldsymbol{Q}^r$	Set of eligible working routes for each demand relation r, indexed by q.

Parameters

Δ	A large positive constant (100000).
c_j^m	Cost of one module of the m^{th} capacity on span j.
z^m	Capacity of one module of type m.
d^r	Number of demand units for relation r.
$\zeta_j^{r,q}$	Equal to 1 if working route q (for demand relation r) crosses span j, 0 otherwise.
$x_i^{p,L}$	Equal to 1 if the L side of cycle p offers an acceptable protection path for failure of span i, 0 otherwise.
$x_i^{p,R}$	Equal to 1 if the R side of cycle p offers an acceptable protection path for failure of span i, 0 otherwise.
π_j^p	Equal to 1 if cycle p crosses span j, 0 otherwise.

Variables

η_j^m	Number of capacity modules of type m placed on span j.
w_j	Working capacity placed on span j.
s_j	Spare capacity placed on span j.
$g^{r,q}$	Demand quantity from relation r that uses route q.
n^p	Number of unit-capacity copies of cycle p in the solution.
n_i^p	Number of copies of cycle p used to protect span i.

$n_i^{p,L}$ Number of copies of cycle *p* required for protection of span *i*, when the L side of the cycle is used.

$n_i^{p,R}$ Number of copies of cycle *p* required for protection of span *i*, when the R side of the cycle is used.

Objective Function

Minimize: $\sum_{\forall m\in Z}\sum_{\forall j\in S} c_j^m \cdot \eta_j^m$ (19)

(Minimize total cost of capacity modules placed.)

Constraints

$$\sum_{\forall q\in Q^r} g^{r,q} \ge d^r \qquad \forall r\in \boldsymbol{D} \tag{20}$$

(All demands must be routed.)

$$\sum_{\forall r\in D}\sum_{\forall q\in Q^r} \zeta_j^{r,q}\cdot g^{r,q} \le w_j \qquad \forall j\in \boldsymbol{S} \tag{21}$$

(Place enough working capacity to support the demand.)

$$\sum_{\forall p\in P}\left(x_i^{p,L}\cdot n_i^{p,L} + x_i^{p,R}\cdot n_i^{p,R}\right) \ge w_i \qquad \forall i\in \boldsymbol{S} \tag{22}$$

(Place enough *p*-cycles, considering L and R paths separately, to protect all working capacity.)

$$\sum_{\forall p\in P}\pi_j^p\cdot n^p \le s_j \qquad \forall j\in \boldsymbol{S} \tag{23}$$

(Place enough spare capacity to build all *p*-cycles.)

$$\sum_{\forall m\in Z} z^m\cdot \eta_j^m \ge w_j + s_j \qquad \forall j\in \boldsymbol{S} \tag{24}$$

(Place enough modules for working and spare capacity.)

$$n_i^p \ge n_i^{p,L} \qquad \forall i\in \boldsymbol{S}, \forall p\in \boldsymbol{P} \tag{25}$$

(Must have more copies than number of paths on L side, for each failed span.)

$$n_i^p \ge n_i^{p,R} \qquad \forall i\in \boldsymbol{S}, \forall p\in \boldsymbol{P} \tag{26}$$

(Must have more copies than number of paths on R side, for each failed span.)

$$n^p \ge n_i^p \quad \forall i \in S, \forall p \in P \tag{27}$$

(Number of copies of *p*-cycle *p* must be greater than the maximum number required by any one failure.)

$$n_i^{p,L} \le \Delta \cdot x_i^{p,L} \qquad \forall i \in S, \forall p \in P \tag{28}$$

(Use no copies of cycle *p* for protection of span *i* if the L side path is unacceptable.)

$$n_i^{p,R} \le \Delta \cdot x_i^{p,R} \qquad \forall i \in S, \forall p \in P \tag{29}$$

(Use no copies of cycle *p* for protection of span *i* if the R side path is unacceptable.)

All variables are constrained to be non-negative integers. Equation (22) ensures that a sufficient number of *p*-cycles are provisioned to protect against all span failures, but uses the new $x_i^{p,L}$ and $x_i^{p,R}$ parameters to ensure this is done without having to use protection paths longer than the hop limit. The two parameters simply define the eligibility of either side of a cycle to protect span *i*. If cycle *p* can provide a protection path on the L side that is shorter than the hop limit (specified during pre-processing), then $x_i^{p,L} = 1$, otherwise it is set to 0. The R side parameter is of course defined in the same way. It is now clear that even if present in the network graph, a cycle may not qualify to protect one of its straddlers if it cannot offer a sufficiently short protection path. A similar logic exists for on-cycle spans as well.

Equation (24) adds modularity (as in [GrDoLEOS02]). This is optional but helps model the problem as closely as possible to the real world. Equations (25), (26), and (27) ensure that the number of copies of *p*-cycle *p* in the solution is the maximum of the number forced by any single span failure. Equations (28) and (29) are "backup" constraints to ensure that, if cycle *p* is not eligible to restore span *i* using either the L or R side, then it will not be considered for protection of that span. While not strictly required, (28) and (29) are "added valid knowledge" constraints that provide extra information and may thus hasten the solution process.

6.3. Experimental Setup

6.3.1. Test Networks

Four test networks are used initially for the full joint modular formulation in 6.2.1 above. We also use a larger network for a specific test to be explained later. Three of the four networks are random pseudo-transport networks we synthesized and the last is the NSFNET from [DoGrJSAC00]. The set of demands between node pairs are assigned from a uniform random distribution between 1 and 10 units. Span costs per unit capacity are assigned based on the geographic distance between the nodes as shown in Figure 6-3 and Figure 6-4.

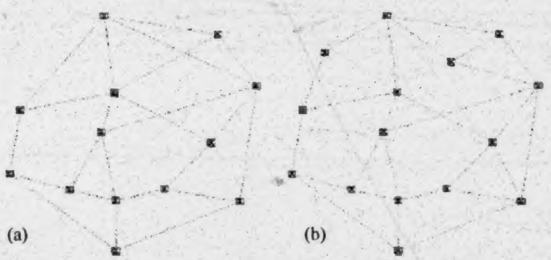


Figure 6-3. Test networks (a) 13n23s (501 cycles), (b) 15n26s1 (871 cycles). (Adapted from [KoSaBROADNETS04])

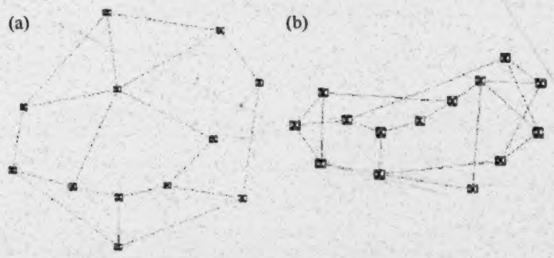


Figure 6-4. Test networks (a) 12n19s (127 cycles), (b) NSFNET (139 cycles). (Adapted from [KoSaBROADNETS04])

6.3.2. Common Parameters

All the ILP models are written in AMPL™ and solved using CPLEX® on a Sunfire V480 server with 16 G of RAM to within 1.2% optimality. The main test cases were solved in a modular environment with two modular sizes – OC-12 and OC-48 to choose from, with a 4x2x economy of scale model. For the hop-limited *p*-cycle test cases, all working routes shorter or equal in length to the tenth successively longest distinct route (by distance) were represented in the input file. For corresponding mesh reference solutions, the same set of eligible working routes was provided. Also for the mesh solution, for each possible span failure, the set of all restoration routes shorter than or equal in length to the tenth longest route was used. The hop limit assertions for the mesh reference designs were built into the model. All the *p*-cycle design problems were given the complete set of all distinct cycles as input (study specific variations are discussed later). The total number of cycles in each graph are given in the captions of Figure 6-3 and Figure 6-4.

6.3.3. Mesh-Restorable Reference Designs

Baseline reference mesh designs were generated with identical hop limit progression as the *p*-cycle test cases, for comparison. To design these mesh solutions, we use the Herzberg mesh SCP design model from [HeByTON95]. The set of all routes is provided to the mesh solver and hop limit restrictions are built into the model with addition of the following constraints to the standard span restorable mesh network design model.

$$\gamma_i^u \leq \Delta \cdot f_i^u \qquad \forall i \in \boldsymbol{S}, \forall u \in \boldsymbol{U}_i \tag{30}$$

$$\gamma_i^u \geq \nabla \cdot f_i^u \qquad \forall i \in \boldsymbol{S}, \forall u \in \boldsymbol{U}_i \tag{31}$$

$$\gamma_i^u \in \{0,1\} \qquad \forall i \in \boldsymbol{S}, \forall u \in \boldsymbol{U}_i \tag{32}$$

$$\sum_{\forall j \in \boldsymbol{S}, j \neq i} \partial_{i,j}^u \cdot \gamma_i^u \leq H \qquad \forall i \in \boldsymbol{S}, \forall u \in \boldsymbol{U}_i \tag{33}$$

The set of eligible restoration routes for each span *i* is $\boldsymbol{U}_i$. The flow over route *u* for failure of span *i* is represented by f_i^u (a variable already present in the standard mesh design model). The first two added constraints force a "one" on the new binary variable γ_i^u as an indication of whether or not a particular eligible route is used for failure of a certain span. The final constraint (where *j* represents any surviving span) uses the also pre-existing 1/0 parameter $\partial_{i,j}^u$, which encodes whether or not restoration route *u* (for failure span *i*) crosses surviving span *j*, to in effect trace out each *used* restoration route and assert that it respect the hop limit *H*.

6.4. Results

6.4.1. Threshold Hop-Limit Effect

Let us now compare the two models: explicit hop limitations on *p*-cycles (HL), circumference limited (CL) *p*-cycles with the baseline, identically hop limited mesh design.

We start with comparing the three designs without any hop limits – i.e. all distinct routes and cycles in the graph are enumerated and admitted to the solver. As a model validation step, the HL and CL designs must be identical, when no hop limits or restrictions are imposed. This is observed on all the test networks and thus validates our model. Second, the mesh, HL and CL must be either very close or equal in efficiency in the unlimited hop environment. Any differences should be to the advantage of the mesh design – or in other words, given strict path length restrictions, *p*-cycles should never outperform mesh designs. What we see is that within a 1.2% MIPGAP, HL, CL and mesh had identical performance. This is therefore consistent with what has been observed consistently in the literature on *p*-cycles.

We then start systematically constraining the hop limit and observe the changes in the performance of HL, CL and Mesh designs at the same corresponding hop limits. The results are plotted in Figure 6-5. At the extreme right of Figure 6-5, corresponding with the 'u' on the horizontal access are the various points for the unlimited test cases we discussed so far. We can now see that as expected, when the hop limit is progressively reduced, all the mesh reference designs exhibit the threshold hop limit effect. For example the 15n26s1 test cases exhibits a threshold of about six hops for the mesh design. The corresponding hop limit for the *p*-cycle HL designs is at about eight or nine. Also, below the threshold, the cost of the *p*-cycle design seems to rise faster than the cost of the corresponding mesh design.

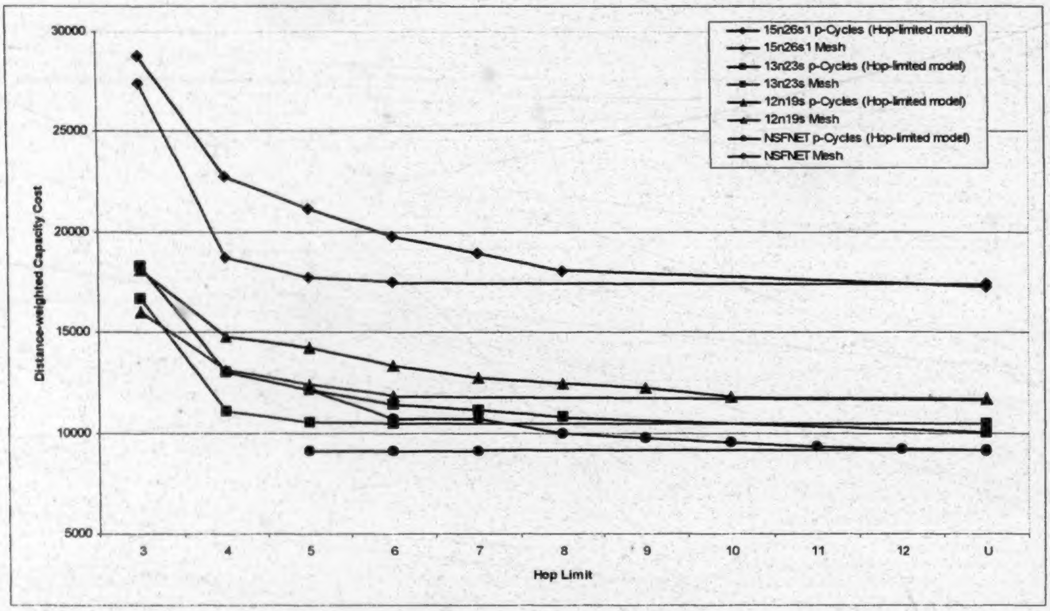


Figure 6-5. Total cost of capacity versus hop limit for hop-limited mesh and *p*-cycle network designs. (Adapted from [KoSaBROADNETS04])

At hop limits as low as 3, the NSFNET design is not even feasible. Those that are have very high-capacity requirements, and the HL, CL and mesh designs almost approach each other. This is as expected as well, because when the solution space is so highly restricted (as the case is at the H=3 point), both mesh and *p*-cycles almost have identically constrained spaces to work with. In the curves for the 12n19s test network, the threshold hop limit is at about six for mesh and nine for *p*-cycles. The increase in costs below the threshold hop limits is also similar to the 15n26s1 test case.

These results in Figure 6-5 show that like mesh, *p*-cycles also exhibit a threshold hop limit effect. The hop limit in a *p*-cycle network design is at about 3 hops higher than the corresponding mesh design. What is important and significant is that as long as both *p*-cycles and mesh designs are both above their respective threshold, their capacity efficiencies are similar. An added concern from these results is that as the hop limits are constrained, *p*-cycle design costs rise faster than the corresponding mesh design costs. At very low hop limits, as might be imposed in a very sparse LHN, both *p*-cycles and mesh approach each other in design cost.

These results give us an overall understanding of *p*-cycles and mesh networks. Until now, the benefits of *p*-cycles were that they combined the best of ring and mesh-based networking principles. Now we know that the compromise is ultimately made when *p*-cycles exhibit a higher hop limit threshold vis-à-vis mesh networks but gain fully pre-connected protection paths. This tradeoff though quite acceptable is what distinguishes *p*-cycles from mesh in capacity efficiency. The difference in threshold hop limits is a network specific detail but for our test networks we find that to be between 3 and 4. The most important new insight is that it is not just in unlimited hop designs that *p*-cycles and mesh match each other in capacity efficiency. As long as both designs are above their corresponding thresholds, *p*-cycles are as good as mesh.

6.4.2. Comparing To Circumference-Limited Designs

Let us now look at how the HL designs compare to the CL designs. In order to achieve a path length limit *H,* the CL design can only use cycles that are smaller than $H+1$ hops. HL designs can include cycles of any size but may only use protection paths that are a maximum of *H* hops in length. Figure 6-6 is the distribution of the restoration path lengths in HL designs for network 13n23s with $H=6$ compared to CL designs with $C=7$. The HL designs have fewer of the longer paths, and more number of short paths, but this overall effect is quite small – only 2.5% of the protection paths are shortened in this way. Additionally, for networks considered for the results in Figure 6-5, the cost of the *H* hop HL designs was the same as the $C=H+1$ CL designs. On detailed inspection of the HL designs, we could see that the solution did include, in the case examined, at least one cycle that was larger than the prescribed hop limit, but at the same time was used to provide protection paths of acceptable length to straddling spans. In other words, the HL designs do sometimes avail themselves of the potential benefits of the explicit hop limiting strategy as discussed in 6.1.6. This is demonstrated by the existence of a 8-hop cycle in one of the $H=6$ HL solution. Evidently that cycle is so efficient at protecting the straddlers that the solver chose to build it despite the fact that it cannot be used to protect any on-cycle spans. But overall, this is a very rare scenario and in most of the cases, the easier CL method is an effective surrogate for strict HL, when $C=H+1$. The small difference in the average path length distribution combined with the rarity of the effect described in 6.1.6 actually occurring in a test case poses an interesting question. Does this difference significantly improve the solution in larger test networks? Or is CL an effective and simple surrogate for HL? Answering this question would be of interest to both us and the network planner. In the test cases in Figure 6-5, we found that while the CL designs produced results in a matter of seconds, the corresponding HL designs took hours and as long as a day to run to termination within the 1.2% MIPGAP termination condition. If CL is indeed an effective surrogate for HL, then the network planner can solve the faster and simpler CL design with the confidence that there is no capacity penalty in not using the relatively more complex HL approach.

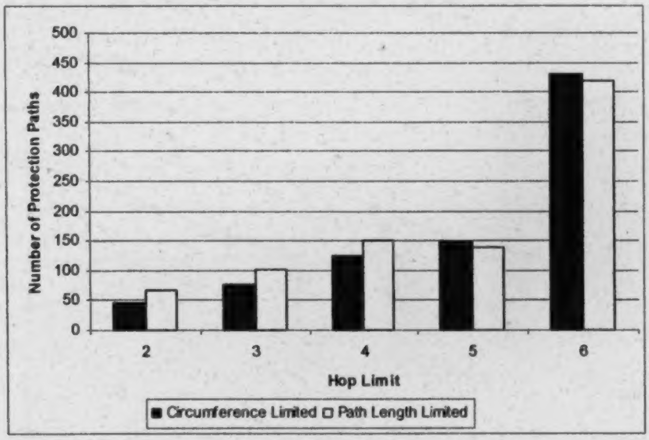


Figure 6-6. Hop-limited path distribution for *C*=7 and *H*=6 designs in the 13n23s network. (Adapted from [KoSaBROADNETS04])

6.4.3. Non-Joint, Non-Modular Design

As a further check on this conclusion, we ran another set of tests using a larger 19-node 35-span test case (from [KoSaBROADNETS04]). In addition, to ensure that modularity and economy of scale were not somehow obscuring any new insights, we ran the large test case as a pure spare-capacity-placement problem without modularity or jointness. The results for this run are in Figure 6-7. Figure 6-7 again validates our previous claims because we find that the capacity requirements for CL and HL designs are identical (within a 1% MIPGAP) at all hop limits. The characteristic hop limit for this network for the *p*-cycle design occurs at about 7 hops higher than the mesh design. Figure 6-8 shows the distribution of path lengths for the 19n35s network test case. The apparent similarity of Figure 6-8 and Figure 6-6 simply confirms our conclusions that CL results in a path length distribution that is not significantly different than the HL solution, which is considerably more complicated. Thus not only does CL produce results that are close (or identical) to the HL solutions, but the details of the solution, such as the distribution of path lengths is also consistent.

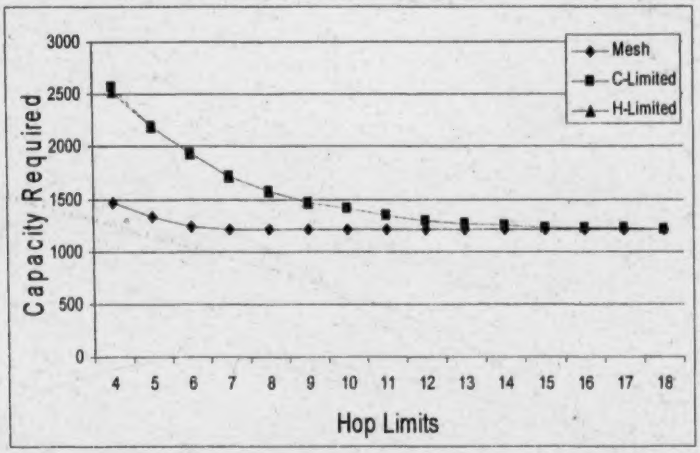


Figure 6-7. Capacity results for the 19n35s network. (Adapted from [KoSaBROADNETS04])

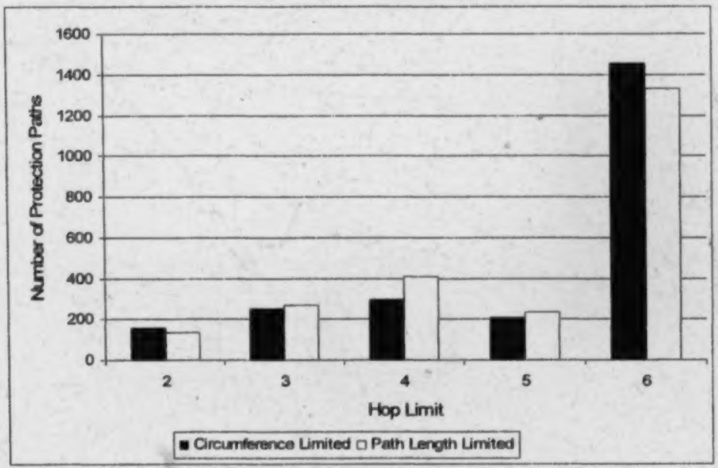


Figure 6-8. Hop-limited path distribution for *C=7* and *H=6* designs in the 19n35s network. (Adapted from [KoSaBROADNETS04])

6.4.4. Bi-Criteria Techniques

After seeing the rather small improvement provided by the HL over CL in the path length distributions in Figure 6-8 and Figure 6-6, we realized that it could be because the solver may not be able to fully 'load' all the *p*-cycles. That is to say the protection potential of the various *p*-cycles may not have been fully exploited. To verify this, and to see what benefit there may be in including some element of choice in the choice of shorter protection paths. We decided to modify the objective function in (19) to a bi-criteria[xvi] objective function as shown below in (34).

$$\text{Minimize: } \sum_{\forall j \in S} s_j + \varepsilon \cdot \sum_{\forall j \in S} \sum_{\forall p \in P} \left(\phi_i^{p,\mathrm{L}} \cdot n_i^{p,\mathrm{L}} + \phi_i^{p,\mathrm{R}} \cdot n_i^{p,\mathrm{R}} \right) \tag{34}$$

(Minimize spare capacity cost and total hop length of all paths used.)

The $\phi_i^{p,\mathrm{L}}$ and $\phi_i^{p,\mathrm{R}}$ values are simply parameters that specify the length of the protection paths, in hops, on either the L or R side of every *p*-cycle *p* for failure of span *i*. These values were obtained as part of the pre-processing step. Of interest is to see how much this bi-criteria objective improves the distribution of path lengths. The results for this experiment are in Figure 6-9. In Figure 6-9 we can see that further improvement is obtained by biasing the HL results towards using shorter paths. The bi-criteria weight (ε) is set to 0.0001. In the CL design, the average path length was 5.08 hops whereas with the bi-criteria technique it dropped to 4.77. The figure also shows a significant jump in the number of two hop paths while overall the number of the relatively long 6 hop paths decreases. If this decrease eliminates the need to place even four or five regenerators in a large network, the cost saving in capital and operational expenses may be in the millions.

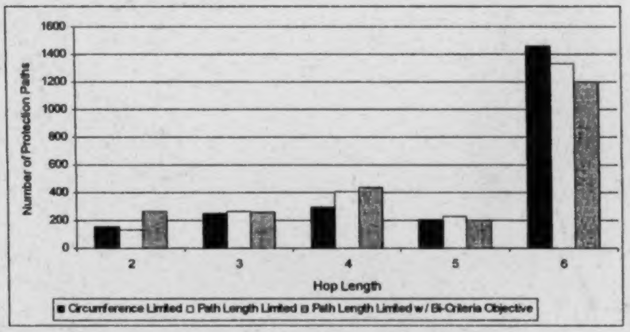


Figure 6-9. Effect of bi-criteria objective in improving path-length-limited results for 19n35s design (C=7, H=6). (Adapted from [KoSaOSN05])

6.5. Summary

In this Chapter, we investigated the issues related to path-length-limited *p*-cycle design. One of the contributions of the work is a design model that can precisely restrict path lengths in *p*-cycle design. With this model we were able to prove the existence of a threshold hop limit effect in *p*-cycle network design. We also found that as long as *p*-cycles and mesh designs are above their corresponding thresholds, they are identical in capacity efficiency. This means that the fully pre-connectedness of *p*-cycles is essentially at no-cost. This model also finally addresses the previously open research problem of precisely comparing mesh and *p*-cycle networks. Another finding in this study was that the threshold hop limit in *p*-cycle networks was about three or four hops higher than the corresponding mesh design. This therefore establishes the quite acceptable dimension that is traded off to gain *p*-cycle properties such as pre-armed, pre-connected protection and the ability to re-use of ring-equipment as nodal equipment. A bi-criteria version of the basic path length restricting design model was also developed. In this model the solver was biased towards choosing a solution that had, on average, shorter protection paths, but at essentially the same capacity efficiency of the regular design model. Future work may include the application of this model to considering regeneration costs arising from path lengths.

As a final contribution, this work also shows that designing with circumference limits is an effective surrogate to using the more complex hop limiting approach. Though the initial existence-proof type example showed that there is a chance the HL solution can be more capacity efficient, in practice, no such benefits were found. There is also virtually no difference in the path length distributions in the HL and CL designs. A potential caveat in this assumption may occur if regenerator costs are significant or important. In this situation, precise path length limiting may be the way to go.

Chapter 7. MULTIPLE QUALITY OF PROTECTION *P*-CYCLE NETWORK DESIGN

7.1. Introduction

In this Chapter we develop design theory and related operating concepts to support multiple differentiated levels of survivability assurances for individual demands in a transport network that uses *p*-cycles as the basic protection method. Notably this includes an ultra-high availability (dual span failure protected) service class that is based on principles that are unique to *p*-cycles for dual failure protection without dynamic reconfiguration of *p*-cycles. The ideas and methods developed in this Chapter can enhance the cost-competitiveness of transport network operators and enable a range of service offerings to best suit actual customer requirements on transport survivability.

This Chapter is based on a recent publication at IEEE BROADNETS 2005 [KoGrBROADNETS05]. An extended version of this paper has been invited for submission to the IEEE-OSA Journal of Optical Switching and Networking [KoGrOSN06] and is currently in preparation.

7.1.1. Motivation and Goals

In a competitive business with a diverse set of users and applications, it is generally desirable to be able to provide a range of service offerings in some efficient way. Ideally, all differentiated services can be provided using only one common set of resources, configured as needed to match the actual service mix. In recent years this viewpoint has been applied to the levels of survivability assurance (and hence availability guarantees) given to various light-path or OC-n path services through a transport network. In approaching this goal, of providing for multiple Quality of Protection (multi-QoP), one of the most essential aspects of a realistically feasible solution is that all services can be provided for within a single integrated framework for operating and planning such a multi-QoP network. What we mean is that a solution that requires separate network architectures, layers, or different equipment types for each service class is not an attractive solution.

The vision here is of a single transport network based on broadband or optical cross-connects at every node within which up to five (if desired) different service classes can be provisioned as required with each individual service path also being efficiently provided with arrangements to assure its particular service level guarantees. All of the currently promised survivability guarantees, ranging from assured dual-failure survivability down to preemptible unprotected service, will be efficiently and collectively coordinated within one pool of network resources, accessed and managed for both provisioning and survivability by the cross-connects.

Survivability related assurances under a service level agreement (SLA) have come to be called QoP specifications, as an obvious extension of the prior general concept of Quality of Service (QoS) in other dimensions such as BER, cell-loss, delay, etc. Protection mechanisms like BLSR-Rings and 1+1/1:1 APS usually support only two QoP types: protected against single failure and unprotected (i.e. "extra traffic" use of the protection channels). This is because a BLSR is line-switched, so any tributary in the working line rate signal gets protected, whether intended or not. Working traffic on the protection channel is the other class and is in effect the pre-emptible QoP class. (In contrast, UPSRs can support protected, unprotected and extra traffic QoP classes, but are not the usual type of ring used in backbone transport.) While this suited voice traffic, many feel today that it is not as well suited to the variety of present and future Internet traffic types. Data—now the dominant traffic borne by the transport network—is better suited to a range of QoP levels. E-mail traffic, for example, can tolerate outages of several hours before the users notice a failure while Voice over IP (VOIP) remains almost as outage-sensitive as TDM voice. The pressure to better match protection resource investments to the real needs of the traffic flows is also driven by economics. Although brute-force, the still common practice of pure 1+1 diverse duplicated routing at least doubles the cost over the corresponding non-protected service. So differentiation, efficiency, and refinement of the protection arrangements used ought logically to be almost at the top of list of potential productivity enhancements for a network operator. The leverage on being intelligent and sophisticated in the management of survivability can approach a factor of 100% in productivity gain out of a given deployed investment in transport equipment. The emphasis is made even greater by the relatively low revenues from many data transport services. For instance, reportedly 80% of British Telecom's revenue in recent years came from services that used only 20% of its network capacity[xvii] [ART03]. The implication is that for the other 80% of the traffic, the same QoP treatment may not be economically justifiable. It also suggests that perhaps the 20% of traffic that earns 80% of the revenue should be given even better QoP assurances than currently provided.

This inspires one of our current questions of interest: Would it be possible to provide guaranteed dual-failure survivability to the high-revenue services with little or no more resources than the uniformly single-failure protected network requires? Instead of a network in which all traffic inherently receives an assurance of single failure protection (denoted R1=1), the goal would be to use the same total resources to provide an ultra-high availability QoP for the most profitable services, based on assured dual-failure survivability (R2=1), and also provide and an R1=1 and notionally "less than R1=1" service definitions for other classes of transport. It seems reasonable that enabling such multi-QoP service offerings would help a network operator stay competitive in a commoditized market for transport services. We therefore consider the problem of efficiently designing *p*-cycle-based networks to support multi-QoP combinations of demands. Prior work (to be reviewed) has looked at multi-QoP design for span-restorable networks with service classes topping out at R1=1. The important difference with this work is that we solve the corresponding problems for *p*-cycle networks and at the same time we extend the multi-QoP framework to directly include a new R2=1 service class, integrated in the overall design (or configuration) problem. Also novel is that it is through two distinct principles, specific to *p*-cycles, that the R2=1 service class is realized.

7.2. Prior Literature

Concepts of differentiated QoP can be found in the literature starting (to our knowledge) in 1996, when a four-tier QoP class set was proposed for Asynchronous Transport Mode (ATM) networks[VeHa96]. Gerstel and Sasaki adapted and extended this for ring-oriented broadband transport networks in [GeSaOPTICOMM01]. More recently optimal capacity design models for span restorable mesh networks with a mix of QoP types was treated in [GrCIICOCN02]. In [GrCIICOCN02], however, the highest QoP class was 'gold' which had assured protection against any single span failure that affected a gold service path. An interesting and motivating finding of [GrCIICOCN02] was that over a fairly wide range of QoP mixes, which include a preemptible service class, there can be no truly idle 'spare' capacity in the usual sense—*all* channels would be in use bearing some form of revenue-producing service. One interest of the present work is to see if similarly desirable combinations of multi-service demand patterns and capacity configuration arise for *p*-cycles. Also relevant to this work is [StGrICC04] which, separate from a multi-QoP concept, developed ideas about how to support an ultra-high availability (R2=1) service class in a span restorable mesh network that had no more spare capacity than normally otherwise needed for R1=1. The key to this was to have a set of capacity-efficient pre-planned protection responses to a *first* failure, and follow this up in the event of a *second* failure with a dynamic adaptive restoration response in the dual-failure state but aiming then only to restore the R2 class service paths.

7.3. Multi QoP Stack

Let us now briefly re-cap the four multi-QoP service classes from [GrCIICOCN02] which we will now incorporate in *p*-cycle networks, and add to this the new dual-failure protected class as illustrated in Figure 7-1.

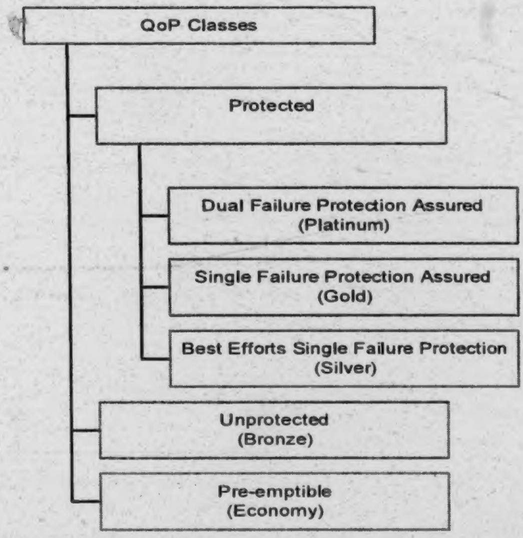

QoP Classes
Protected
Dual Failure Protection Assured (Platinum)
Single Failure Protection Assured (Gold)
Best Efforts Single Failure Protection (Silver)
Unprotected (Bronze)
Pre-emptible (Economy)

Figure 7-1. Multi-QoP class set. (Adapted from [KoGrBROADNETS05])

Working channels of the platinum class are assured protection from all possible dual and single span failures in the network. Channels bearing gold service paths are assured protection against all possible single span failures and now will also receive a best-efforts response to dual-failures using any spare (or preemptible) resources remaining after the platinum services are fully restored. Working channels bearing silver service paths are protected on a "best efforts" basis for any failure scenario (single or dual) that affects such paths. In the event of a single failure, affected silver channels will be restored to the greatest extent possible using any spare channels not employed for protection of gold or platinum services. Here, however, the best-efforts response for silver services does not include the option to preempt economy services (as was considered in one case in [GrClICOCN02]). Any response for silver in the event of a dual failure is also prioritized below platinum and gold service restoration requirements. Bronze working channels are unprotected, but are not preemptible by higher class services either. Working channels bearing economy class services are unprotected and may be pre-empted if required for protection of platinum or gold class working channels. Such preemption occurs, however, only when true spare capacity provided on available *p*-cycles is not adequate to meet guarantees for platinum or gold services. Note that (as in [GrClICOCN02]) from the standpoint purely of network capacity design, there is no real distinction needed between bronze and silver working channels as neither of these classes warrants any designed-in protection capacity for them and they also are not usable for protection of other services. From a capacity design standpoint, therefore, there is nothing to do for them, other than to allow for their own working capacity in the design. The difference is manifested operationally, however. In real time affected silver channels may receive restoration paths, but bronze do not. The preemptible or "Economy" service class is of particular interest. Many BLSR ADMs possess a feature called "Extra Traffic" provides preemptible access to the protection channels of the rings. Some carriers use this feature to transport traffic such as special-event TV signals or to provide additional non-essential or temporary extra logical links between routers. Therefore economy class service paths in a *p*-cycle network conceptually involves forming *p*-cycles out of ordinary "spare" capacity channels, but then allowing the *p*-cycles to be used much like the Extra Traffic feature of rings. Thus, the role of Economy service channels is especially important in the overall multi-QoP design problem because they play three roles that the design model will ideally reflect and exploit. These are: (i) they can best contribute to revenue by being placed in a way that supports routing of the available economy demands, but at the same time they must (ii) also be formed into closed cycles to act as *p*-cycles, and (iii) when used as *p*-cycles, must efficiently provide for gold and platinum restorability guarantees.

7.4. Preplanning and real-time operation of multi-QoP

Here we outline the operational concept for multi QoP networks based on *p*-cycles. Most of the logic follows from [GrClICOCN02] but has to be extended and adapted to specifically recognize the *p*-cycle environment and the additional R2=1 service class. First, each service path (a demand routed over the network between is OD end-nodes) is labelled with a service class tag at provisioning time. Thus, although we speak of channels as being gold or silver, or of other classes, channels are really all intrinsically the same and only temporarily inherit the protection status of the service path of which the channel is currently part of. The routing of working multi-QoP labelled demands in the network generates a specific number of working channels on each span *i* of each service class. This is the number of the individual platinum paths through the span w_i^p, gold working paths over the span, w_i^g, and so on where w_i^{sb} will denote silver and bronze working capacity as an aggregated group and w_i^e denotes the number of economy channels on span *i*. These are all integral multiples of some unit granularity working channel for example STS-1 units in a SONET-managed network, or light-paths in an optical transport layer. Each working channel on a span is labeled according to the class of protection of the service path it is currently serving. Using simple protocols this state can be acquired and maintained locally by the end-nodes of the span. The capacity design process (or the closely related configuration problem for existing capacities) will then ensure that each gold or platinum channel is protectable by specific *p*-cycles as needed. This information is stored locally at the nodes adjacent to each span, as a simple lookup table. The information is not globally required and is specific only to the incident spans at each node. Thus, in the pre-failure network state, each node knows of all incident working channels, and their individual protection statuses, and knows the inventory of all *p*-cycles accessible at the node.

In real time, the only further considerations are that once damaged or in use, the other nodes of any *p*-cycle see that status advertised in the overhead bytes on the signal the cycle is bearing. With these provisions, the implementation of apparently complex multi-QoP policies is actually very fast, simple, and completely local. At all times, there is simply a ranked matching problem to implement. Whether it is a single or dual failure scenario does not even complicate things, because in all cases the end nodes of any failed channel simply match the available protection cycles to the failed working channels in rank order among platinum, gold and silver, while completely ignoring bronze in the list of failed working channels, and completely ignoring economy services riding on the available *p*-cycles at the node. The end-nodes of the failed span then only need to detect whether a particular working channel needs and warrants protection, or not. If the channel needs protection, they must simply look up a local data table and re-direct the affected traffic into one of the available *p*-cycles. The adequacy of the protection capacity and currently configured *p*-cycles to meet all QoP guarantees is an aspect of the off-line design problem, not the real time activation problem. When a service is routed and provisioned, these checks are made within the current capacity design and configuration. If need be, incremental re-optimization of the *p*-cycle set can be employed to support the

QoP assurances. Ultimately, when a capacity-limited situation is reached, it means that (as in any network), until capacity augmentation can be realized, new SLAs for affected QoP classes cannot be issued. The important thing overall is that this combination of off-line capacity design (or capacity configuration) and the locally-applied real time mechanism described, means that QoP assurances, once given for a newly provisioned service path, really are guaranteed, and the reaction of the network in real time is always simple and locally reacting. Additionally this is done without the use of separate layers to support demands of various QoP classes.

7.5. Dual Failure Protection in *p*-Cycle Networks

We now review some prior methods used for dual failure protection in *p*-cycles and then propose two simple principles for dual failure protection.

7.5.1. Relationship to Prior Work

Schupke et al. [ScGrICC04] first proposed a mechanism for enhanced dual failure survivability in *p*-cycle networks. In Schupke's work, cycles of all sizes are allowed but the maximum number of protection relationships that each cycle can provide, is restricted. This is called reducing the susceptibility of the network to dual failure. This reduces the number of spans that are exposed to a dual span failure. Schupke also proposed the use of dynamic re-organization of *p*-cycles after the first failure as a possible solution. The strategy is to re-organize the spare capacity on the surviving spans into new *p*-cycles that maximize the restorability of the reduced network. The re-organization was either global, where all *p*-cycles were re-built, or incremental where existing *p*-cycles are kept untouched, but at the same time an attempt is made to assign more protection relationships to existing *p*-cycles. This re-organization is operationally done through either a centralized or distributed run-time re-optimization or a pre-calculated local reaction to a specific span failure. Option (*i*) would entail a complete real-time, re-optimization of the capacitated sub-graph (with the failed span and in-use protection resources excluded), while the second approach involves continuously running a series of planning problems, which produce the net switching actions necessary at every node in case of a failure of any single span. In effect the network operations center continuously rehearses what it has to do in the event of a failure, to maximize readiness to subsequent failure. Results in [ScGrICC04] show that global re-organization of *p*-cycles following the first failure can achieve 100% restorability to dual failure with about 70% more capacity than what is required to ensure single failure survivability. Note that this implies that the network could be as much as 300% redundant. Another conclusion from [ScGrICC04] is that no amount of over-provisioning can guarantee 100% dual failure survivability using only the susceptibility minimization strategy.

In this work, we do not rely on dynamically reconfiguring the *p*-cycles after the first failure or on explicit additions of extra spare capacity. Our novel approach emphasizes dual-failure survivability achieved through an initial design of static *p*-cycles, and the integration of a dual-failure survivability service class into an overall multi-QoP framework. Insights from [ScGrICC04] and [ClGrPNET03] about the two basic ways in which *p*-cycles can withstand dual failures are acknowledged and relied upon here, however. To our knowledge this is the first work in general on optimized capacity design to consider an dual failure survivable (denoted R2=1 or just R2) service class integrated into an overall environment of five QoP classes, all sharing the same physical capacity. With more certainty it is thought to be the first study to do so specifically for a survivable network based on *p*-cycles. What is also different from the past dual-failure restorable design work on span-restorable networks is that while [ClGrPNET03] was based on the unique properties of a span-restorable mesh network, the attainment of a R2 service class in *p*-cycle networks here is based on quite different principles that are specific to *p*-cycles. It is not just a reapplication of concepts for achieving R2 in span-restorable networks.

We think the main contributions are: (i) development of a *p*-cycle multi-QoP capacity design model, (ii) elaboration and integration of two basic principles for dual failure protection in *p*-cycle designs (other than post-failure reconfiguration) into the muti-QoP design framework, and (iii) providing numerical results on the network implications of the multi-QoP design model.

In the next section we describe two simple principles for dual failure protection that do not require any reconfiguration. These principles can be used in an essentially 'static' *p*-cycle network to provide full dual failure protection to platinum class demands. The DPP principle can easily be extended to triple or '*n*' failure protection, without reconfiguration. These principles are based on the realization that if two diverse protection options are provided per unit of platinum capacity on a failed span, then regardless of the exact combinations of spans that fail, dual failure protection can be guaranteed.

7.5.2. Straddling Span Protection Principle

One principle we can exploit for dual failure protection is based on simply not to fully load the straddling spans with two protected paths of the top priority on a straddling span. A *p*-cycle always offers two diversely routed protection paths per unit of its own capacity to each straddling span it protects. Normally these are used to protect two different working paths on the straddling span. But, conceptually, if a platinum path crosses a straddler, nothing prevents both of these diverse protection paths from being kept available for one platinum path. Thus before a failure, there are three disjoint routes between the two end nodes of the protected span. This means that any two failures affecting the service can be survived.

Note, importantly, however, that for efficiency in design we can still use *both* protection paths in the case of single failures for two different working paths. All that we must ensure is that the service using the protection path that was not used by the platinum service in a first failure is itself of lower status than platinum. This way design capacity efficiency is every bit as efficient as it is for normal single failure survivable (denoted R1=1 or just R1) design alone, but when a second failure arises (affecting the same platinum service in this *p*-cycle), the latter is in a position to exploit the *p*-cycle's other surviving protection path, implicitly bumping any lower-status service that was initially using the other protection path following the first failure. Figure 7-2(a) shows a *p*-cycle that protects span *AB* as a straddling span. This *p*-cycle offers two disjoint protection routes to span *AB* shown by the two arrow-tipped thin lines. In Figure 7-2(b) span *AB* fails. The gold channel on span *AB* is protected using the route shown by the dashed line and platinum channel is protected using the route along the dotted line. Figure 7-2(c) then shows a second span failure on span *CD* that affects the protection path of the platinum channel on span *AB*. Immediately, the end-nodes *A* and *B* reuse the protection path that previously protected the gold channel to now protect the platinum channel from *AB* instead. Note that this all happens automatically and reflexively using only node-local information because the second failure is simply an event that causes nodes *A* and *B* to detect a loss of light on the backup path. This triggers an update to the node-local problem of priority matching with available protection paths. There is no extra signaling needed at nodes *C* or *D* (the end nodes of the second failed span) for the protection of platinum channels on span *AB*. When the second failure hits, the set of available protection paths is reduced, a platinum path is seen exposed at the top of the priority list, and so being higher in ranking, it is applied to the path gold previously enjoyed.

Note the overall principle and implications: *If a platinum demand is routed entirely over straddling spans it is protected end-to-end against dual span failures* (as long as it is not accompanied by another platinum path straddling the same unit-capacity *p*-cycle). Furthermore, in a multi-QoP environment, *the dual-failure protection for a certain number of platinum demands can be completely without additional spare capacity requirements above an R1=1 design* because the same *p*-cycle that offers dual-failure protection to platinum path on one of its straddlers, remains available for routing of single-failure protected paths over the same straddler.

Note conversely that straddling span routing is also the *only* way that single *p*-cycles can provide dual-failure protection. By single *p*-cycles we mean that the dual-failure protection is provided entirely within the scope of one *p*-cycle per span along the path, without involving dispersal of failures over multiple *p*-cycles (which follows). The reason is that in a *p*-cycle, the only other possibility is that the path being protected has one or more on-cycle spans of the *p*-cycle. In this case a *p*-cycle offers only one disjoint protection path against a failure and therefore cannot be the source of dual-failure protection. [xviii]

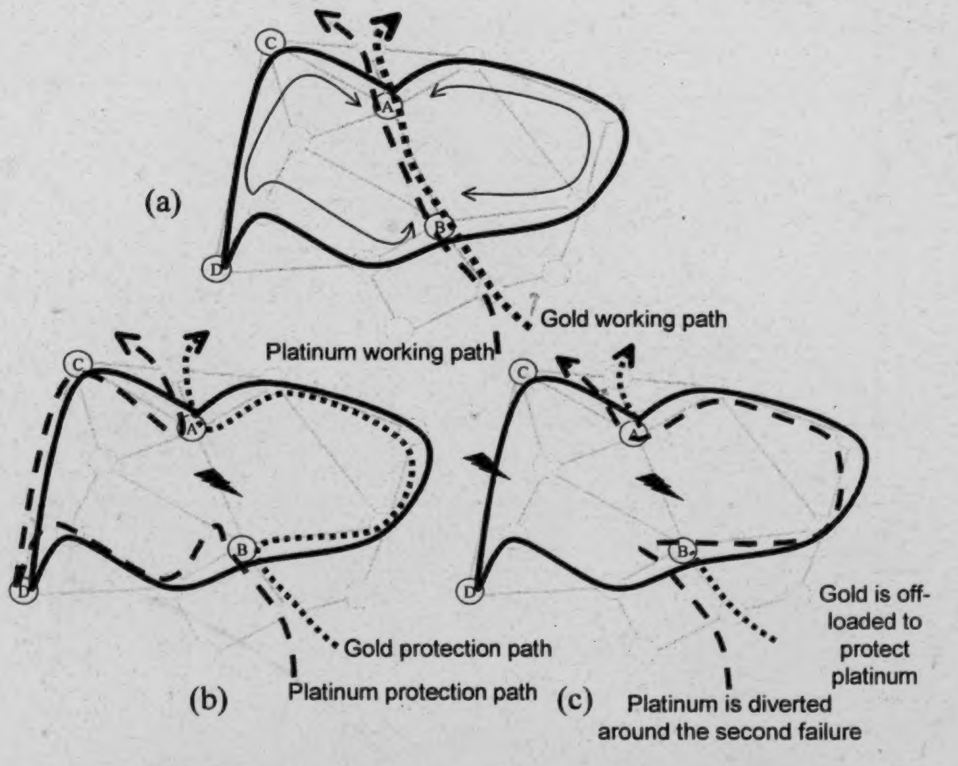

(a)
C
A
B
D
Gold working path
Platinum working path
C
A
B
D
Gold protection path
Platinum protection path
(b)
C
A
B
D
Gold is off-loaded to protect platinum
Platinum is diverted around the second failure
(c)

Figure 7-2. Straddling span routing principle for dual-failure protection (a) *p*-cycle *X*, (b) first span failure, (c) second span failure. (Adapted from [KoGrBROADNETS05])

7.5.3. Dispersal Protection Principle

The principle of failure *dispersal*, somewhat similar to the fault dispersal principle from [ScGrICC04], is another way to arrange for assured dual-failure protection of platinum services, without necessarily incurring any capacity penalty whatever for incorporation of the desired level of single-failure restorability, The principle is similar to straddler related dual-failure survivability in a single *p*-cycle, but it is more general in that we effectively disperse the two failures into two different *p*-cycles which are fully or partially disjoint. For example, if a span is in a topologically straddling relationship to two *p*-cycles that are mutually span disjoint, then there are actually four disjoint protection routes for a working channel on that straddler. In all, there are three general cases: (i) A span *AB* straddles *p*-cycles *X* and *Y* and *X* and *Y* are span disjoint. This provides four fully disjoint topological routes for protection. (ii) A span *AB* straddles *X* and is "on-cycle" to *Y*, and *X* and *Y* are disjoint. This yields three disjoint topological route options. (iii) A span *AB* is on-cycle to both *X* and *Y* but everywhere else *X* and *Y* are disjoint: then two disjoint protection routes exist. Figure 7-3 gives an example of dual-failure dispersal over a pair of *p*-cycles with the type (ii) relationship. discussed above.

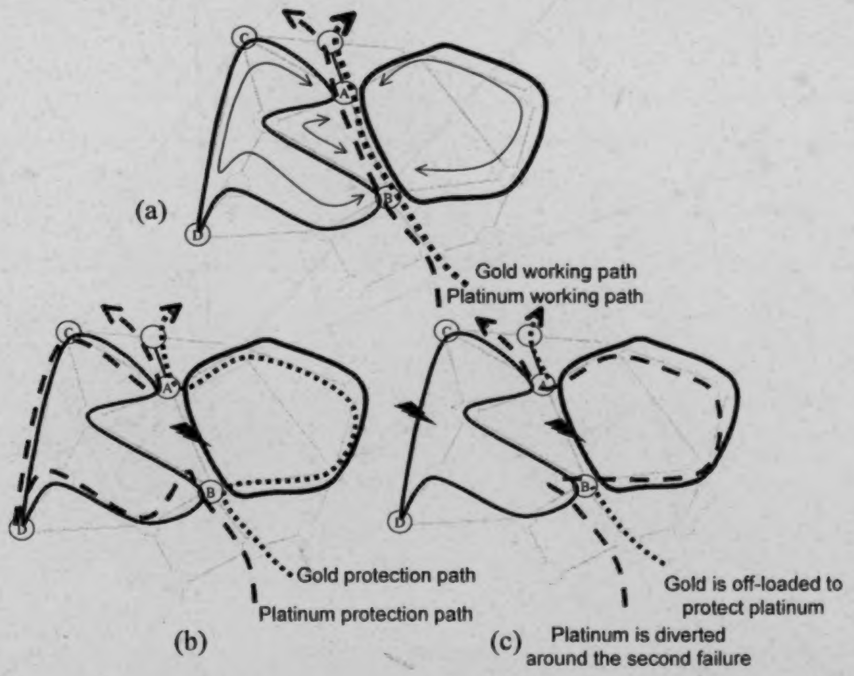

(a)
C
D
A
B
Gold working path
Platinum working path
(b)
Gold protection path
Platinum protection path
(c)
Gold is off-loaded to protect platinum
Platinum is diverted around the second failure

Figure 7-3. The dispersal principle for dual-failure survivability (a) *p*-cycles X and Y, (b) first failure and (c) second failure. (Adapted from [KoGrBROADNETS05])

In Figure 7-3(a), span *AB* is a straddler to *p*-cycle *X* (the cycle on the left) and is "on-cycle" to *Y*. Thus together, *X* and *Y* can offer three span disjoint protection options to each working channel on span *AB*, along the routes indicated by the think arrow-tipped lines in Figure 7-3(a). Figure 7-3(b) shows a failure scenario in which span *AB* has failed. We then choose to protect one platinum channel on *p*-cycle *X* using the route indicated by the dashed line and one gold channel using *p*-cycle *Y* using the route indicated by the dotted line. Figure 7-3(c) then shows the second failure of span *CD* which is on the protection path of the platinum channel (on *p*-cycle *X*). The end-nodes *A* and *B* bump off the gold channel from its existing protection path and allocate it to the platinum channel. Alternatively the nodes can use the other available protection route on *p*-cycle *X* (shown in Figure 7-3(a)) and thus manage to protect both the platinum and gold working channels simultaneously.

7.6. ILP-based Design

In this section, we bring all of the prior concepts and considerations (except the dispersal principle) together in an optimization-based model for the overall design on multi-QoP *p*-cycle protected networks that include the new dual-failure protected platinum service class. The purpose of the model is to permit research investigations of the properties and potential capabilities, benefits, and the study of networking science of networks of this type. The model itself, or heuristics based on insights from study of the optimum model, may later be used in practical network planning and related studies for QoP related business strategy, and so on.

The model that follows is of a capacity-design orientation. In other words, it takes given requirements in each QoP class and generates routing, capacity placement, and *p*-cycle decisions to serve and protect all demands according to their QoP class at minimum total cost. From this it is not hard to derive corresponding models where the total capacities are initially given and other objectives, such as total demand served, or total revenue, are maximized subject to available capacity. The model also incorporates modularity of capacity and economy-of-scale [DoGrCCECE99].

As mentioned, the present model fully incorporates the straddler-based routing principle but the dispersal principle in design is not presently included. Doing so will only increase the efficiency and effectiveness of the resulting designs, however. In closing this section, we will therefore discuss the issues surrounding the addition of the dispersal principle to the optimal design model and outline our approach to obtain these added benefits in ongoing work.

Sets

S	Set of spans, indexed by i_1 (first failed span) and i_2 (second failed span) and j (an surviving span).
X	Set of eligible cycles, indexed by p.

D^p	Set of Platinum demand requirements, indexed by *r*.
D^g	Set of Gold demand requirements, indexed by *r*.
D^{sb}	Set of Silver & Bronze requirements (merged), indexed by *r*.
D^e	Set of Economy demand requirements, indexed by *r*.
Q^r	Set of eligible working routes for each demand relation *r*, indexed by *q*.
M	Set of available capacity modules.

Parameters

Δ	A large positive constant.
ρ_i^x	Equal to 1 if span *i* is on cycle *x*. Equal to 2 if span *i* straddles cycle *x*, 0 otherwise.
π_j^x	Equal to 1 if cycle *x* crosses span *j*, 0 otherwise.
d^r	Total number of demand units exchanged by the r^{th} node-pair (of all QoP classes).
$\zeta_j^{r,q}$	Equal to 1 if eligible working route *q* (for demand relation *r*) crosses span *j*, 0 otherwise.
z^m	Capacity of m^{th} module type.
C^m	Cost per unit distance of module type *m*. (The values in these coefficients can reflect a non-linear economy-of-scale in capacity versus relative cost.)
L_j	Length of span *j*.

Variables

w_j^p	Platinum working capacity placed on span *j*.
w_j^g	Gold working capacity placed on span *j*.
w_j^{sb}	Silver and Bronze working capacity placed on span *j*.

w_j^e	Economy working capacity placed on span *j*.
$g^{r,q}$	Quantity of demand from relation *r* that uses q^{th} route for that demand relation.
s_j	Spare capacity placed on span *j*.
t_j^m	Number of modules of type *m* placed on span *j*.
n^x	Number of unit-capacity *p*-cycles to form on candidate cycle *x*.
$n_i^{x,g}$	Number of unit-capacity instances of cycle *x* used to protect gold working capacity on span *i*.
$n_i^{x,p}$	Number of copies of cycle *x* used to protect platinum working capacity on span *i*.

Objective Function:

Multi-QoP-JCP: Minimize: $\sum_{m\in M}\sum_{j\in S} t_j^m \cdot C^m \cdot L_j$ (35)

(Minimize total cost of capacity modules placed.)

Constraints

$$\sum_{q\in Q^r} g^{r,q} = d^r \quad \forall r \in D^p \cup D^g \cup D^{sb} \cup D^e \tag{36}$$

(All demands must be routed.)

$$\sum_{r\in D^p}\sum_{q\in Q^r} \zeta_j^{r,q} \cdot g^{r,q} = w_j^p \qquad \forall j \in S \tag{37}$$

(Place enough working capacity to support all Platinum demands.)

$$\sum_{r\in D^g}\sum_{q\in Q^r} \zeta_j^{r,q} \cdot g^{r,q} = w_j^g \qquad \forall j \in S \tag{38}$$

(Working capacity to support all Gold demands.)

$$\sum_{r\in D^{sb}}\sum_{q\in Q^r}\zeta_j^{r,q}\cdot g^{r,q}=w_j^{sb}\qquad \forall j\in S \tag{39}$$

(Working capacity for Silver-Bronze demands.)

$$\sum_{r\in D^{e}}\sum_{q\in Q^r}\zeta_j^{r,q}\cdot g^{r,q}=w_j^{e}\ \forall j\in S \tag{40}$$

(Working capacity for Economy demands.)

$$\sum_{x\in X}\rho_i^x\cdot n_i^{x,g}\geq w_i^g\qquad \forall i\in S \tag{41}$$

(Protection provided by the *p*-cycles must protect all Gold working capacity against single failures.)

$$\sum_{x\in X}\rho_i^x\cdot n_i^{x,p}\geq w_i^p\qquad \forall i\in S \tag{42}$$

(Protection provided by the same set of *p*-cycles must also protect all Platinum capacity against single failure)

$$n^x\geq(n_i^{x,g}+n_i^{x,p})\ \forall i\in S,\forall x\in X \tag{43}$$

(Provision the maximum number of unit-capacity *p*-cycles on cycle *x* required by any span *i* for single-failure protection of both w_i^p and w_i^g.

$$n_i^{x,g}+n_i^{x,p}\geq w_i^p\qquad \forall i\in S,\forall x\in X \tag{44}$$

(The set of *p*-cycles also has capacity for dual-failure protection of platinum services on span *i* including recapture of gold protection.)

$$n_i^{x,p}\leq\Delta\cdot(1-\pi_i^x)\ \forall i\in S\ ,\forall x\in X \tag{45}$$

(Asserts straddler-only routing of platinum paths.)

$$\sum_{x\in X}(\pi_j^p\cdot n^x-w_j^e)\leq s_j\qquad \forall j\in S \tag{46}$$

(Generate spare capacity to build all *p*-cycles allowing preemption of economy capacity.)

$$w_j^p+w_j^g+w_j^{sb}+w_j^e+s_j\leq\sum_{m\in M}t_j^m\cdot z^m\ \forall j\in S,\forall m\in M \tag{47}$$

(Generates modular capacity to support all working and spare capacity channel counts.)

Constraints (36) to (40) ensure that sufficient working capacity is placed on each span for the routing of all demands. Regardless of QoP class, every demand consumes working capacity for its own basic routing. Constraint (41) and (42) ensure that gold and platinum capacity are both protected against single failures and constraints (10) and (11) address the dual-failure survivability of platinum demands. Constraint (10) reflects that to provide dual-failure protection for platinum paths, we can either reuse protection that was built-in for both gold and platinum under single failures or (implicitly) generate additional *p*-cycles and spare capacity as needed for platinum dual-failure protection. Note the potential here for the spare capacity ultimately to be forced by either the two classes together (gold and platinum grouped) under single failures, or platinum alone under dual failures. Results will return to this point. Constraint (11) is the assertion required from principles described above that platinum paths must be routed only over spans with straddling relationships to their protecting *p*-cycles. The constraint works by denying any protection credit for a platinum channel on a span i from a cycle x that contains span i (hence is in an on-cycle relationship). Constraint (46) ensures that spare capacity exists to build all the necessary *p*-cycles with provision to use economy channels as equivalent to spare capacity in constructing *p*-cycles. If one removes w_j^e from this constraint, pre-emption of economy is avoided. Constraint (47) introduces modularity and economy of scale into the problem.

7.7. Experimental Setup

7.7.1. Modeling The Dispersal Principle

As mentioned, in the above model and in the designs in this paper we presently forego the additional opportunities to also derive dual-failure survivability through dispersal of dual failures into more than one *p*-cycle. This means that, as attractive and interesting as the results are which follow, they are actually conservative as to what should ultimately be obtainable in terms of dual-failure protection efficiency. Addition of dispersal considerations in the model is a challenge, however. The central issue is that exact representation of dispersal opportunities for inclusion in the optimal design model involves relationships not just of each candidate cycle to the graph itself (as at present), but of information on all pair-wise relationships of candidate cycles to each prospective span failure which is $O((\#\ of\ cycles)^2\ *\ \#\ spans)$. This makes a formal optimal model of much greater complexity and beyond the scope of the paper. In practice we think that ways to exploit this added opportunity lie in post-inspection of designs from the model above and in on-line routing algorithms that can exploit that added effect once a set of *p*-cycles is already chosen. In addition, it may be reasonable to represent only a selected subset of dispersal pair relationships in a practical ILP model, analogous to the way eligible routes or eligible cycles can be sometimes limited in practice for span-restorable mesh or basic *p*-cycle ILP design problems.

7.7.2. Test Networks

Initially the two small test networks in Figure 7-4(a) and (b) are used to study properties of the multi-QoP network designs. In a later set of tests, the larger test network in Figure 7-4(c) is used. The XnYs naming indicate *X* nodes and *Y* spans. In both of the test cases all nodes are of degree 3 or higher. While this is an essential requirement for full R2=1 designs, it is not essential in a multi-QoP environment but in tests this property allows us to specify platinum demands on any node pair.

The more general requirement is that there must be at least three disjoint routes in the graph between any two nodes that wish to exchange a platinum-class demand. This is a minimum topological requirement for dual-failure survivability by any method whatsoever, and is in no way specific to *p*-cycles. To conduct design experiments where all node pairs exchange a mix of demand types, including one or more platinum path requirements, it is therefore necessary to work with such tri-connected sample graphs.

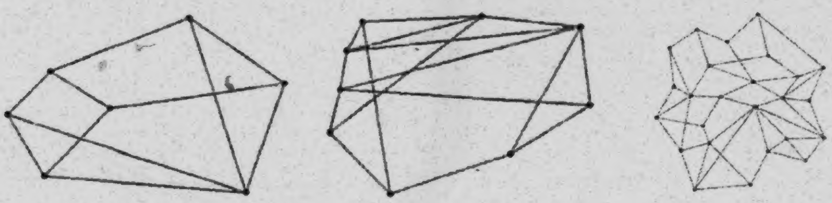


Figure 7-4. (a) 7n11s (b) 9n16s test networks and (c) 25n50s test network. (Adapted from [KoGrBROADNETS05])

7.7.3. Computational Details

In the following design trials, the total number of demands between individual node pairs was assigned from a uniform random distribution between 100 and 200. The demand graph is thus always a fully connected mesh, although with differential capacities per O-D pair. We solved the capacity placement problem for both the modular joint and non-modular non-joint working routing cases. For the joint problem, all routes up to and including the 10th successive shortest eligible route were represented for the working path routing decisions (All eligible routes of length equal to the maximum are included in this set). For the non-joint problem we used the single shortest path for working routing. For both test networks, all possible cycles were included in the candidate cycle set. The model was represented in AMPL 9.0 and solved using CPLEX9.0 on a Windows 2000, dual AMD-Opteron machine with 1 Gigabyte of RAM. Exact run-times were not of concern. However, all the formulations ran to completion with a 1% MIPGAP within one hour.

Of the total demand bundle on each O-D pair, the QoP mix was varied over the experiments, but for repeatability, made proportionally the same for each O-D pair. For example, a 20-60-10-10 solution means that 20% of the total demand on every O-D pair was comprised of platinum, 60% gold, 10% as Silver and Bronze, and 10% as Economy. Fractional values were rounded up, as needed for integrality. Span costs are linearly proportional to the distance between the end nodes of the spans in the network graphs as drawn above and non-linearly proportional to capacity. The available capacity modules were 12 and 48 following a 4x2x economy of scale model.

7.8. Results

7.8.1. Joint Modular Design With Only Gold and Platinum

We first used the model to run a series of tests where we progressively increased the percentage of platinum in a pure platinum-gold QoP mix. The interest in these experiments is to validate and study the exchange between gold and platinum capacity in the design. If the explanations of 7.5 are correct, we should see the predicted effect where, up to some point, the platinum QoP guarantees are met with no extra spare capacity above that required only for single-failure QoP assurances for both gold and platinum. The results in Figure 7-5 for design capacity and cost versus percent platinum confirm this. Design capacity is the total number of logical channels needed for both working and spare capacity of all types. Design cost is the total cost of all capacity modules used to provide the required logical capacity and includes the economy-of-scale effect. Figure 7-5(a) shows that there is no capacity increase required until at least 30% of every demand bundle is platinum. In other words, 30% of all demand in the test case can be protected against all dual span failures with only the capacity budget of the single failure protected design. In other words, it is the single failure considerations for protection of both gold and platinum as a single group with R1 requirements that forces the spare capacity requirements, while above 30% it is the dual failure protection requirement of platinum capacity that alone becomes the forcer of spare capacity. This shift in forcing structure will vary depending on the test case but it is an important general insight about how gold and platinum QoP classes share spare capacity.

The spare capacity profile when increasing platinum demand percentages is in Figure 7-5(b). As in Figure 7-5(a), we find that almost 30% of every demand can be assigned platinum class protection under the total modular spare capacity budget of the all-gold, single failure protected design. We now find that it takes between 1.8 and 2.5 times more spare capacity for the all-platinum design as compared to the all-gold design. The initial dip in total spare capacity results for the 9n16s network in Figure 7-5 (b) arises from the joint and modular nature of the designs. Under the economy of scale a joint, modular, minimum cost design can sometimes choose working longer paths that result in lower cost for the total of both working and spare capacity needed. In a joint design, where the percentage of platinum demands is steadily increased, the spare and working capacities may vary, but the total modular cost and therefore total capacity is always either unchanged or increasing, and this is verified in Figure 7-5(a).

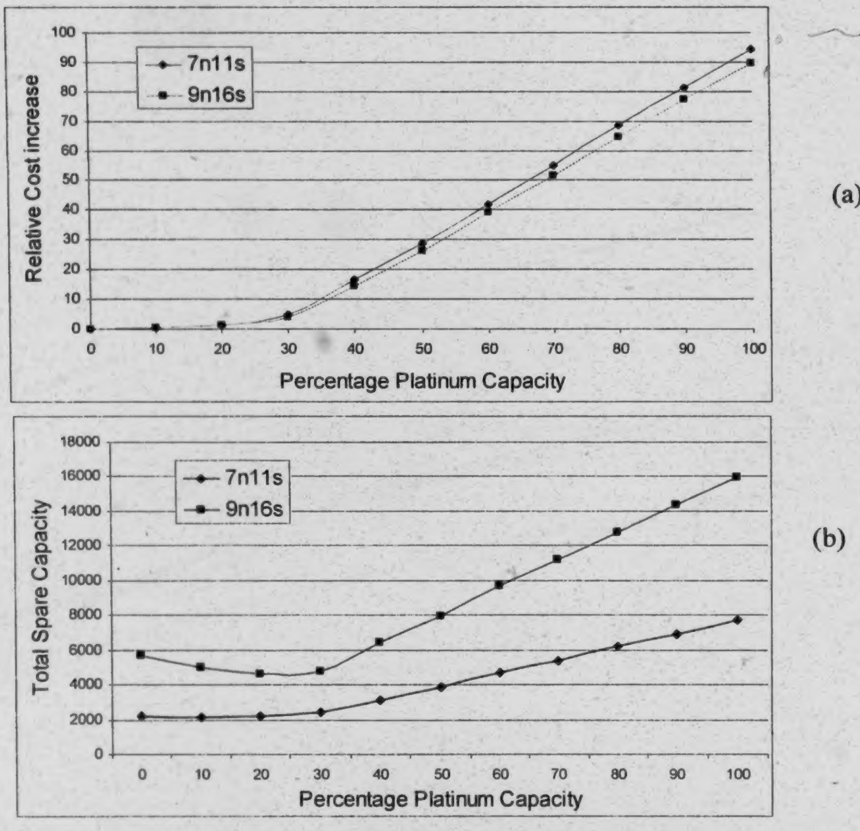


Figure 7-5. Results for the Joint Modular min cost design (a) relative percentage increase and (b) total spare capacity. (Adapted from [KoGrBROADNETS05])

To ensure that modularity economy of scale and jointness effects are, however, not somehow obscuring basic effects such as interactions between QoP classes, we also ran the model in its non-joint, non-modular form on the same test networks. In this case there is only one eligible route for each working path, its shortest route, and (in effect) only one capacity module is available of size equal to one channel. Figure 7-6 shows these results and confirms that the basic effect of up to 30% "free" platinum services is not somehow attributable only to excess channels arising from modular design under economy-of-scale. In the non-modular design, up to 30% of every demand bundle can again enjoy platinum protection without any significant increase in the total spare capacity of the network. Now, however the all-platinum test case takes as much as 2.9 times more capacity than a uniform R1=1 design. This is consistent to what was found before ([GeSaOPTICOMM01][Cl04]) for span restorable mesh network designs that are completely designed for 100% dual span failure. In both cases it underlines that fully R2=1 networks are very costly and that it is both far more sensible, and also quite feasible, to provide R2=1 only for premium services, and to do so in a multi-QoP environment that uses the same set of resources to support all QoP levels simultaneously. The increase over the previous ratio of 2.5 in the joint design is because now the working channels are routed through the shortest path, which is not necessarily the optimal path that the solver would have chosen in the corresponding joint design.

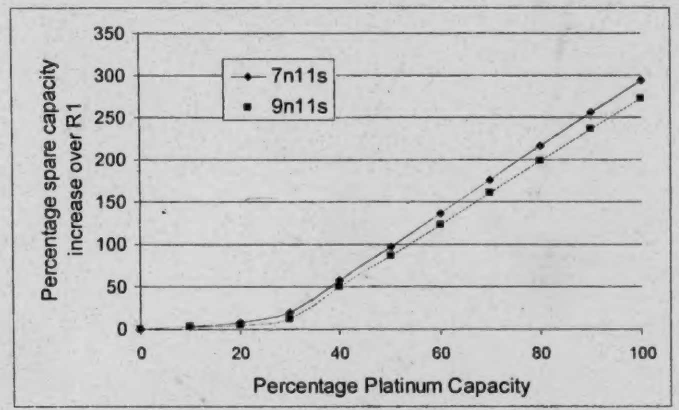


Figure 7-6. Spare capacity requirements for non-joint, non-modular design. (Adapted from [KoGrBROADNETS05])

7.8.2. Experiments on Larger Networks

So far, the results do not involve any economy traffic in the demand mix. Being preemptible for protection of platinum or gold class working channels, the presence of economy class demands can have significant effects and opportunities under the optimal design strategy. Economy channels are unique in that, while they have revenue-bearing services routed through them, they are simultaneously equivalent to 'spare' capacity for gold or platinum services in case of a failure. Ideally, multi-QoP service mixtures can enable the service provider to defer or eliminate additional investment while supporting ongoing growth. In this section, we extend the prior results to a) consider growth characteristics in multi-QoP networks of this type, b) discuss the effects of having preemptible traffic in the mix, and c) to show that a slightly simplified model of the multi-QoP design problem be solved on a medium size network. The test network 25n50s is shown in Figure 7-4(c). Demand bundles of exactly 100 units were established between every node pair. Whereas in the smaller test networks, the set of eligible cycles was the complete set of all distinct cycles of the graph, we now use all cycles up to the 10000th shortest cycle (out of a total possible 616559 cycles in the graph) as the input eligible cycle set in the design problems and results are obtained only for the non-joint non-modular design case within a 1% MIPGAP. The growth-related aspect of these tests involved applying an increasing demand multiplier uniformly to the entire demand matrix for each of several multi-QoP service mix scenarios and re-solving the design problem to observe the rate at which the cost increases for different QoP mixes undergoing growth. Applying a uniform demand multiplier to every demand is one repeatable way of emulating demand growth. More specific growth forecasts can be used in actual planning applications with the same model. Results for the constant demand multiplier as applied to an all-platinum, all-gold, and some reasonable QoP mix test cases are in Figure 7-7. Figure 7-7 is a plot of the absolute spare capacity totals (channel counts) against the multiplier applied to all demands. The QoP mix cases are denoted by their percentages – 10P80G10E means that 10 percent of every demand is assigned platinum protection, and 10% is pre-emptible economy traffic, while the rest is assigned as gold. We do not consider any silver or bronze traffic in the mix for the same reasons as discussed in 7.8.1.

In Figure 7-7 working capacity is not included because in all cases it scales identically with the growth multiplier itself, for obvious reasons. The spare capacity requirements, on the other hand, grow at a rate that depends considerably on the QoP mix, especially with effects such as platinum-gold reuse of spare capacity and gold or platinum preemption of economy channels, all being involved simultaneously in the design. Not surprisingly in Figure 7-7 we see that the all-platinum design spare capacity cost grows at a much faster rate than the all-gold and QoP-mix curves. Table 6 gives absolute and relative slopes for the curves in Figure 7-7. Because the all-gold design represents the conventional R1=1, single failure protected gold class service, it is taken as the relative cost normalizer. These slopes can be interpreted as the rate at which the spare capacity cost increases with growing demand for a particular demand mix, as compared to the corresponding rate of increase in the all-gold design. The all-platinum spare capacity cost grows approximately 3.5 times faster than that for the all-gold design. The all-gold and multi-QoP mix demand matrices facilitate growth at a much lower overall rate of cost increase associated with protection. For example, with 40% preemptible economy traffic, the rate of spare capacity cost increase is only 0.16 times the increase in the all-gold case.

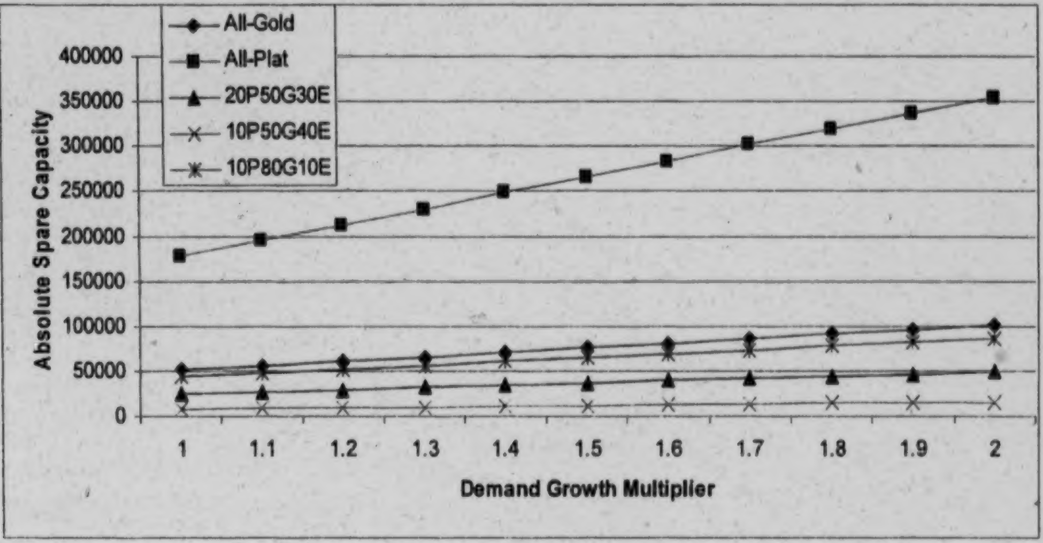


Figure 7-7. Results for 25n50s test case with static QoP mix and Lambda. (Adapted from [KoGrBROADNETS05])

A separate observation from the results on 25n50s is the significant effect of economy demands being present in the QoP mix. For example, in the 10P50G40E case the actual spare capacity required to support the total survivability requirements of the 60% platinum and gold demands is 15% of the total protected capacity. In other words, these two subsets of demand enjoy protection with only 15% redundancy, because the design exploits the opportunities to also preempt economy channels. In such a network, almost every channel generates some revenue.

Table 6. Absolute and Relative rates of spare capacity growth versus demand growth factor for different multi-QoP mixtures. (Adapted from [KoGrBROADNETS05])

QoP Mix	Absolute (Channels/0.1 growth)	Relative
All-Platinum	17731	3.5
All-Gold	5066	1
20P50G30E	2458.6	0.49
10P50G40E	807.96	0.16
10P80G10E	4321.1	0.85

7.9. Summary

We have shown how a multi-QoP framework, including a dual-failure protected service class, can be efficiently implemented in a static *p*-cycle network. To our knowledge this is the first application of multi-QoP design to *p*-cycle networks and the first report of multi-QoP design involving not just "gold-and-below" QoP classes, but also efficient integrated provisioning for a dual-failure survivable "platinum" service class in the same QoP stack. Results indicate that with the straddling span routing principle for meeting dual failure requirements, significant fractions of dual-failure withstanding platinum services can be supported with no more total capacity than is otherwise required for uniform single-failure restorable design. The results also show that the introduction of economy demands can be very significant. Over a wide range of multi-QoP mixtures, survivable network services can be provided with absolutely no truly idle spare capacity. This work also confirmed that the spare capacity requirement for 100% dual-failure design with *p*-cycles is about three times that for 100% single-failure design, similar to prior findings for span–restorable mesh networks. Future work can consider methods to incorporate the dispersal principle into the network design for even greater efficiency and related models to maximally load existing-capacity networks with multi-QoP demand matrices. The latter models should not be difficult extensions from the central capacity-design multi-QoP model given here and can provide avenues to approach the problems of either incremental re-optimization of multi-QoP networks and/or completely dynamic provisioning algorithms for multi-QoP random arrivals and departures.

Chapter 8. FAILURE INDEPENDENT PATH PROTECTING *P*-CYCLE NETWORKS

The author and his supervisor recently proposed a new technique for optical network protection called Failure Independent Path-Protecting (FIPP) *p*-Cycles. The method is based on an extension of *p*-cycle concepts to retain the property of full pre-connection of protection paths while adding the property of end-to-end failure-independent path protection switching against either span or node failures. In Chapter 4 we saw that in a translucent or transparent optical network, until all connections are made, it is not actually known if the backup optical path will have adequate transmission integrity. FIPP *p*-cycles support failure-independent, end-node activated switching like SBPP but *with* the fully pre-connected protection path property of *p*-cycles. As a fully pre-connected and path-oriented scheme, FIPP *p*-cycles are therefore potentially more attractive for optical networks than SBPP. In this Chapter we present four design models for FIPP *p*-cycle networks.

This work on Failure Independent Path Protecting *p*-Cycles is based on four published papers. The first work on FIPP *p*-cycles is published in the IEEE Journal of Lightwave Technology [KoGrJLT] followed by an invited paper at the International Conference on Transparent Optical Networks, [GrKoICTON], a conference paper at DRCN 2005 (in collaboration with Dr. John Doucette) [KoGrDRCN05], and a journal submission to the OSA Photonic Network Communication [KoBaPNET06]. An in-preparation submission to Photonic Network Communications, in collaboration with Dimitri Baloukov [KoBaPNET06] is summarized in 8.4.3.1 with corresponding results presented in 8.8. A US patent on FIPP *p*-cycles has been filed [GrKoUSPT] and summarizes the invention.

8.1. Introduction

As we saw in 2.10, the main features of the *p*-cycle concept are that it combines the properties of ring-like speed with the capacity-efficiency of a span-restorable mesh network. The practical importance of the *p*-cycle property of providing *fully pre-*

connected protection paths in optical networks was given recent emphasis in work motivated towards achieving this property with linear "pre-cross-connected trails" (PXTs) as protection structures in [ChChTON04]. Pre-cross-connection, PXTs and other related topics are reviewed in considerable detail in section 4.5. In general there will be a significant speed advantage if the protection structures are fully pre-cross-connected, compared to a network where protection paths are cross-connected in real-time on an on-demand basis. This is the main source of the speed that rings have always enjoyed, and it was one of the main motivations for the original development of *p*-cycles. But in an optical network, as discussed in section 4.4 and 4.5, pre-connection can be even more important than just being a speed-related issue because when optical protection paths are pre-connected, they can be pre-engineered and tested, and be in a known working condition (vis-à-vis power and data rate budgets, loss, dispersion, etc.) prior to their use. This is important, as with current optical networking and switching technology, it is not realistic to expect that an on-the-fly concatenation of arbitrarily selected spare wavelength channels cross-connected between different spans at the optical level will result in an end-to-end path with under 10^{-12} bit error rate (BER). Without relying on opaque (electronic core) cross-connects to in-effect cross-connect the payloads not the optical channels themselves, it remains a difficult problem to arbitrarily connect several optical channels directly with assurances of immediate end-to-end transmission quality at 10 Gb/s to 40 Gb/s in a DWDM environment. This is often overlooked in the literature on dynamically switched DWDM optical networks but we are told by industry colleagues that it is one of the biggest practical stumbling blocks for network operators that would like to consider dynamic mesh restoration or shared backup path protection using transparent optical cross-connects. Therefore, full pre-connection of protection paths is a very important property because the paths can be engineered and tested before failure and will then be known to work with certainty when accessed for restoration.

In this Chapter, we describe the concept of FIPP *p*-cycles, as an extension of the basic *p*-cycle concept into a path-oriented scheme. From the results of the design models herein, we find that the most important properties of FIPP *p*-cycles appear to be that they remain competitive or at least comparable in capacity-efficiency with SBPP (and other path-oriented restorability mechanisms), and like SBPP provide simple failure-independent end-node activation and control against either span or node failure, but they employ only fully pre-connected protection paths. In effect, they provide fully pre-connected protection paths at no capacity penalty with SBPP, the current state-of-the-art in path oriented survivability. To explain FIPP *p*-cycles let us first review some relevant background that is specific to FIPP *p*-cycles and therefore not included in Chapters 1 to 4.

8.2. Background

8.2.1. Path Protection, Restoration and Mutual Capacity

In path-oriented protection or restoration in general, the objective is to provide (or to form on demand) a surviving backup path for each affected working path in the event of any single span or node failure that disrupts the path. Path-oriented survivability schemes are generally more capacity efficient than their span-based counterparts and end-to-end re-routing gives customers control on activating the backup path(s) for their affected services. Although adaptations or extensions of span-protecting schemes have been developed to cope with node failure as well as span failure (see [DoGrNFOEC03] and [KaReDRCN03] and [StaGrJSAC00] for example), path-based schemes have a more inherent ability to react against either a span or node failure. On the other hand, most path-oriented survivability schemes are slower and more complex than span-oriented schemes. Notably, in the sense that the word "protection," as it originated in Automatic Protection Switching (APS) and ring applications, really implies the existence of a fully pre-cross-connected bi-directional backup path between end nodes, it seems to us that no *shared*-mesh *path*-oriented *protection* scheme has strictly yet been developed. By this we mean a scheme that is:

a. capable of being designed for single failure protection of 100% of the network demands if desired,

b. has spare capacity requirements that are *very* much less than 100% redundancy—close to the theoretical minimum of spare capacity set by optimal multi-commodity maximum flow solutions for restoration,

c. provides strictly and simply pre-cross-connected bi-directional protection paths between end nodes, and,

d. requires only end-node failure detection (no immediate fault localization is needed), and,

e. only the end nodes need do any protection switching.

Other factors aside when comparing schemes, it is also generally agreed that a preferred protection scheme will always be one that requires less network state information and state-update dissemination (for robustness) and fewer logical constructs to create, change and monitor to sustain network survivability goals (for op-ex reduction), especially as demand patterns evolve. In considering the background literature, first, of course, we recognize that diverse dedicated 1+1 APS has a protection path that goes end-to-end and is fully pre-connected. But the cost of this is at least 100% redundancy. There is no "shared mesh" aspect of efficiencies from reusing protection capacities over multiple different failure scenarios. In contrast, Shared Backup Path Protection (SBPP) has very good capacity efficiency, and is end-to-end oriented, but it is not a protection scheme (in the sense above). It is really a pre-planned restoration scheme, which has no aspect of backup path pre-cross-connection. The routes of backup paths are decided in advance, but a path must be formed on-demand by seizing and cross-connecting spare channels on that route when needed. More precisely, we would say that SBPP is a failure-independent pre-planned path restoration scheme. However, SBPP does stand as the most closely related and relevant scheme against which to compare FIPP *p*-cycles, and we will be returning to consider it in further detail.

Another scheme that needs to be considered under the criteria above is "demand-wise" shared protection [KoZyJNSM05]. The concept (in its most general form) is to split the total bundle of demand between any two end-nodes over multiple mutually disjoint routes of the graph and also use those disjoint routes to support an end-to-end allocation of protection paths for each working route. Backup paths assigned to protect one working route may be co-routed with the other working routes, or routed along separate routes. For instance, if three mutually disjoint routes exist between nodes A and B, and the total demand between A and B is 11 light-paths, then the best solution is to assign $w_i = \{0,6,5\}$ working paths, respectively, and $s_i = \{6,0,0\}$, $i=1,2,3$ spare paths. In this example, the path-wise logical redundancy would be 6/(11) = 54%. The true ratio of total capacity used above that required simply for shortest path routing of demands (the "standard redundancy")[xix] would be higher than this however, because in general, the disjoint paths are not shortest paths. The overall capacity efficiency is thus limited by the fact that sets of mutually disjoint routes tend necessarily to include routes that are much longer than the shortest route and by the fact that the minimum edge-cut between most node pairs in typical transport networks is only two or three. Between any two nodes with a min-cut of 2 edges, 100% is the best redundancy achievable. With a min-cut of three, it is logically 66%, but total capacity cost will not reflect this full benefit because of the excess routing lengths of the three disjoint routes involved. Ultimately these effects result in there being only 2.6% and 8.7% capacity cost reduction *relative to 1+1 APS* in the test case results for protection of all demands in [KoZaJNSM05]. Therefore, while the "demand-wise" shared protection scheme would have the other desired properties, it does not have the characteristically high-capacity efficiency that characterizes a "shared mesh" solution.

Finally, in searching the literature for schemes that might combine all of the key properties above, we also found the approach of backup light trees as developed in [ShYaPNET04][[GrPNET05]. The basic idea involves adoption of a unidirectional tree-forming approach in the "incoming" direction on each node, as a destination node. Each individual tree is pre-configured out of a single channel of protection capacity and arranged so that some (not all) of the light-paths incident on that node can enjoy a disjoint route from their origin to the respective destination node through the particular backup tree for that node in that direction. Multiple distinct unit-capacity backup trees are defined on each destination node until all demand flows in that direction to that node are made single-failure restorable. There are several issues when lined up for comparison to the criteria above. One is that the backup trees are pre-configured but not pre-cross-connected in the hard and simple sense that pre-connection is unusually meant, such as in rings, *p*-cycles or 1+1 APS. Rather, specific assumptions about incoming signal mutual exclusion, valid signal recognition, and signal selection at merge nodes on the tree, and at the destination node, are required to support the use of the backup tree concept on nodes where pre-connection of degree > 2 is required in the trees. The arrangements for protection are also not bi-directional. Each direction of each working demand will wind up taking a different restored-status route over the network. In addition, capacity efficiency is evidently also not comparable to that of SBPP or even span restoration as indicative of what is expected from mesh sharing of spare capacity: [GrPNET05] reports that "results indicate the capacity savings due to (backup trees instead of 1+1 APS) are about 15% for networks with reasonably high degree (about 3.5) and large demand matrices suitable to the topology." In contrast it is already well known to expect that under "shared mesh" architectures, such as SBPP or even a span restoration on a degree 3.5 network, we can conservatively insist achieving something like 50% absolute standard redundancy, or less. (15% better than dedicated 1+1 APS is still a very much higher than this level and likely often still over 100%.)

Failure independence is one of the attractive and simplifying properties that both FIPP *p*-cycles and SBPP share and will also be discussed further. Protection trees or backup trees [StGrTR99][ShYaPNET04] can be fully pre-connected structures but do not operate on an end-to-end basis as in [GrPNET05b][GrPNET05]. They act as span-protection technologies in [StGrTR99][ShYaPNET04]. In the method of [ShYaPNET04] there always remains a requirement for on-demand restoration path assembly to address failures on spans *on* the single backup tree itself. [ShYaPNET04] even mentions that because of rules prohibiting nodes from routing protection through one of its own child nodes, the scheme is also not always able to support 100% restorability. The earlier backup tree scheme in [StMSC97] has no such limitation on restorability, and has good efficiency even with 100% fully pre-connected restorability, but it does operate on a span-protection basis. Other schemes, such as path restoration, span restoration, PXTs (reviewed in 2.12) and so on all omit at least one of the three main properties we now seek. To reiterate, these are:

i) Capacity efficiencies well under 100% and characteristic of optimized SBPP (or path restorable) network designs.

ii) Path-oriented failure-independent end-node activation and control in response to either span or node failures.

iii) Completely pre-connected end-to-end protection paths.

The most efficient scheme that is theoretically possible for survivability (without global rearrangement of unaffected paths) is failure-specific path restoration as considered in [IrMaTON98]. Here, after the failure, a centrally controlled, or distributed adaptive process produces a coordinated set of restoration paths that restore each affected path on all affected end-node pairs in an efficient way. In this type of restoration, the composite set of restoration paths deployed considers the exact location of the failure and is allowed to reuse any working capacity on the failed paths up to and leading away from the failure location. This is referred to as "stub release." It makes path restoration inherently more efficient than SBPP, but also not "failure-independent" in its response, because failure localization is essential for stub-release. Such failure specific path restoration is, however, the most capacity-efficient form of restoration possible when the restoration path sets are computed by (or are equivalent to) a capacitated multi-commodity maximum flow (MCMF) type of optimized routing model. However, to implement such a scheme assumes real-time dissemination of the failure information, and therefore is more complex than schemes for span restoration or for failure-independent path end-node controlled schemes such as SBPP. As such true path restoration tends often to be used only in theoretical studies in estimating the lower bounds of spare-capacity requirement for any possible restoration scheme.

From a theoretical standpoint, the central issue that makes any approach to path restoration (or protection) more complex than for span restoration is the coupling of multi-commodity routing decisions under the finite spare capacities available on spans. This theoretically complicating mutual capacity issue in path-oriented survivability (with 100% restorability guarantees) is discussed at length in [IrMaTON98] [KaGrGLOBECOM03] [Grov03]. For present purposes, we can summarize the issue (discussed previously in 2.9.2) with an excerpt from [Grov03]:

> *"Consider a number of simultaneously affected O-D pairs and imagine for argument's sake that they each took a turn in sequence to obtain restoration paths. One O-D pair may have 20 equivalently good route choices that will restore it, but for another O-D pair there may be*

> *only one particular route on which it can obtain restoration paths. What if the first O-D pair chooses a route that uses up the spare capacity on the only possible route for the second O-D pair? More generally, how can the exact route choices made by each of a multitude of O-D pairs be coordinated so that the spare capacity that is crucial for restoration of one O-D pair is not blindly used up by another, which may have had different routing options in any case. This is called the mutual capacity problem in reference to the central quandary of this situation: to which pairs should we allocate the spare capacity of a span among the many that try to seize it?* "([Grov03] p.383)

The relevance here is that all path-oriented schemes must have some way of directly or indirectly addressing the central issue of mutual-capacity coordination. One place where the issue is, however, ignored is under recent proposals for GMPLS-based independent end-node "mass redial" as a restoration mechanism [KaGrGLOBECOM03]. The price for ignoring the mutual capacity issue is that any such restoration scheme is then inherently only a "best-efforts" approach. There can be no assured outcomes or protection guarantees given by design. For any assured outcome from a path restoration process, the mutual capacity issue must be addressed in one way or another. It is therefore important to the present work to understand how schemes other than GMPLS mass-redial [KaGrBLOBECOM03] do address the issue. SBPP addresses the mutual capacity issue indirectly and elegantly in a pre-failure way by permitting only working paths that are mutually disjoint (more specifically, have no shared-risk link groups (SRLG) in common) to share spare channels on other spans that are in their respective backup routes. Failure-specific path restoration addresses the problem more directly, but with considerable added complexity, through solution of a capacity optimization problem during the design phase that involves Multi Commodity Max Flow (MCMF) type routing subproblems within its overall structure and use of on-line MCMF routing decisions in realtime or an equivalent distributed adaptive restoration algorithm that senses conflicts in capacity allocation among paths for different end-node pairs being simultaneously restored [IrGrDRCN98]. In both cases, however, the mutual capacity issue leads to considerable complexities for path-oriented survivability schemes, either in the advance planning (SBPP) or in the real-time phases (MCMF-like dynamic restoration). With the aim of being path-oriented, FIPP *p*-cycles will ultimately also have to deal with the mutual capacity issue. One prior attempt to extend *p*-cycles to a path-oriented form [ShGrJSAC03],[ShGrICC03],[ShGrCAN03][GrShUS03] led to segment-protecting *p*-cycles (reviewed later) which ultimately coped with the issue by being failure specific in response and adopting an MCMF-type resolution of the mutual capacity allocation problem during design time. In contrast, the key to FIPP *p*-cycles will be to use an SBPP-like disjointness restriction on the working paths that can share spare channels in their protection paths. But unlike SBPP we will retain the fully pre-connected path protected aspect of *p*-cycles.

As discussed in 2.11, SBPP is a pre-planned path restoration scheme primarily developed by the Internet Engineering Task Force (IETF) [KiKoIETF01] for use under Internet-style signaling protocols for protection of light-paths in optical networks. SBPP is, however, logically identical to the ATM Backup Virtual Path (VP) scheme that preceded it [GrZhDRCN98]. Under SBPP one backup route is predefined for each working path and no matter what fails on the working path, restoration is via a path assembled on-demand over this one predetermined backup route. Conceptually, SBPP is like 1+1 APS where two fully disjoint routes, a working and a backup, are established for each signal but for efficiency in the use of spare capacity, we can share spare channels over the backup routes for different working paths. For this reason SBPP is also sometimes described as 1:1 APS with "backup multiplexing." The working paths are usually called "primary" paths. Usually, one or more backup routes are possible between the same end-nodes of the primary path, but only one is chosen for the final design. To be eligible as a backup route, a route has to have no nodes or spans in common with the route of the primary path itself *and* no spans or nodes in common with any other primary path whose backup route has any spans in common with the route being considered. Together these considerations ensure that when a primary path fails (under any single failure scenario) no span or node along its backup route is simultaneously affected. This means it will be possible to assemble a backup path along that route if sufficient spare channels have been pre-planned.

Note in Figure 2-14 that it is individual spare *channels* that SBPP is organized to share and that it is not possible to have these channels cross-connected in advance of failure because it is not known which of the specific backup paths in Figure 2-14(c) might be needed until failure actually occurs. Ultimately, it is because SBPP sharing is structured on a per-channel basis, over groups of mutually disjoint primaries, that SBPP requires cross-connection in real time to form actual restoration paths. (A key point to follow will be that under FIPP *p*-cycles, such corresponding sets of primaries share entire pre-connected structures, not individual channels.)

Optimization models for SBPP design are available in [DoClONM03] and developed in more depth in [Grov03]. More often, however heuristic methods are used for SBPP network design such as for example [OuICC00] and [XiXuJLWT03]. The emphasis on heuristics for SBPP is partly because of the difficulty of solving the optimal SBPP design model even when the complete set of demands is given at once. Heuristics are also better suited to incremental survivable routing which is a strong practical orientation for the SBPP approach. "Incremental" in this context refers to the problem of routing new demands individually as they arrive and arranging the shared-capacity backup path for each arrival in the context of all other already present demands and backup paths. Variations on this type of heuristic can either assume given capacities and try to perform survivable routing so as to minimize blocking, or, to treat each of a set of demands in sequence, attempting to route each one so that the least additional new capacity has to be added and thereby approximate an optimal solution for collective routing and minimum total spare capacity placement.

The key ideas for routing under SBPP are that one tries to route the working path over the shortest or least cost path over the graph while also considering the possible backup routes (disjoint from the chosen working path) and their potential for sharing of spare capacity. On an incremental arrival basis, the optimal incremental SBPP setup is one where the sum of capacity used for the working path plus new spare channels that had to be provided to allow for the required backup path is minimal. It is because the backup route is fully disjoint from the route of the corresponding primary path that the switching action is failure independent. On the other hand, SBPP has some drawbacks. One of these is the extensive database dependencies discussed in [ShGrKLUWER04] and

[ShGrAPOC04] arising from the need in every node for global capacity, topology, and backup-sharing relationships to support dynamic provisioning with SBPP. In addition, SBPP depends on real-time assembly of an actual backup path, which implies signaling, length dependence of the restoration time,[xx] and as argued above, uncertainty about optical path integrity for the dynamically assembled backup path.

8.2.2. Flow *p*-Cycles, a First Attempt

More closely related to the present effort was past work on path-segment protecting *p*-cycles [ShGrJSAC03]. Flow *p*-cycles are reviewed in detail in 2.10.1. Flow *p*-cycles significantly extended the ability of *p*-cycle methods to include node-failure protection (aside from the separate method of node-encircling *p*-cycles) and it also gave a significant further increase in spare capacity efficiency over regular span-protecting *p*-cycles. The key idea in flow *p*-cycles is to address the mutual capacity constraints by relaxing the end-to-end protection requirement to allowing protection of arbitrarily defined path segments. This improves on the spare capacity efficiency over span *p*-cycles but the operational aspects are complicated by failure specificity and the simplicity of strict end to end switchover to a predefined backup path is not achieved. The main complexity of the effort on flow *p*-cycles remained the struggle with the "mutual capacity" issue. As a result, flow *p*-cycles neither obtain failure independence, nor do they incorporate the end-to-end path switching properties that are the current goal.

Another recent development to which FIPP *p*-cycles will be relevant is the concept of a "protected working capacity envelope" (PWCE) for simplified dynamic provisioning of protected services [ShGrAPOC04] [ShGrKLUWER04]. Under the PWCE concept one provisions service paths over inherently protected capacity, as opposed to explicitly provisioning protection for every service. When we route the service through the available channels of a PWCE it is inherently protected. Provisioning protected services through the envelope looks the same as point-to-point routing over a non-protected network. One does not have to make any explicit arrangements for protection of every individual path or globally update network state for every individual path setup (or takedown). Under PWCE, as developed for span-based survivability schemes, once the graph and the vector of spare channel quantities on the network spans are given, there is a unique maximum number of protected working channels available on each span. Thus, a given distribution of spare capacity on a graph creates a uniquely determinable envelope of protected working capacity on each span. Within this operational envelope a vast number of simultaneous service path combinations are feasible and all are inherently protected. Provisioning of a new protected service path is then only a matter of routing over the shortest path through the envelope using only spans that currently have one or more protected working channels available. FIPP *p*-cycles lend themselves to PWCE-type operation as well, but in an even more desirable way because entire routes between O-D node pairs will become pre-defined structurally protected entities.

8.3. FIPP *p*-Cycles

Before we describe FIPP *p*-cycles, let us work through a wish-list of features desirable in an ideal survivability mechanism.

8.3.1. Transport Network Survivability Technology "Wish-List"

Table 7 presents a comparison of all transport network survivability mechanisms discussed so far. At the far right side is a column with the properties of an ideal architecture. Ideally the survivability mechanism must have good capacity efficiency, i.e. far less than 100% redundancy, must be simple to operate and manage and must have assured restoration outcomes. In addition, it is big plus if the individual protection paths are known and tested prior to being used and only the end-nodes of the failed end-to-end light-path need to react.

FIPP *p*-cycles, as we shall see later, are in the same range of efficiency as Path Restoration, guarantee that the individual protection paths actually work, simple to operate and design and only involve switching by the end-nodes. In the following section, we describe the FIPP *p*-cycle concept.

Table 7. Ideal Transport Network Wish list. (Adapted from [GrKoUSPT])

Attribute	Span Restoration	Path Restoration	*p*-Cycles	Flow *P*-cycles	SBPP	p-Trees	PXTs	IDEAL
Capacity Efficiency	good	best	good	~best	~best	bad	(not well characterized yet)	**Good**
Individual protection path known in advance ?	no	no	YES	YES but failure specific	YES	no	YES	**YES**
Operational Complexity	moderate	high	LOW	high	high	high	high	**Low**
Design complexity	low	high	low	high	high	high	very high	**Low**
Assured restoration outcomes	yes	yes(with MCMF-like PR)	YES	YES	YES	YES	~YES	**YES**
Only two Nodes act (per path restored)	no	no	YES	YES	NO	YES	YES	**YES**
Only End nodes act	no	no	no	no	no	no	no	**YES**

8.3.2. FIPP Concept

Although it has eluded ourselves and others until recently, it has occurred to us that there is one potentially very simple principle through which ordinary span-protecting *p*-cycles can be extended to provide an end-to-end path protection technique without floundering on the operational complexity of the mutual capacity and failure specificity issues of flow *p*-cycles. The key is not to try to find the *p*-cycle equivalent to failure specific path restoration per-se but to ask instead: What is the *p*-cycle equivalent of a failure-independent path-protection scheme such as SBPP. Implicit in this is willingness to settle for less than the best possible capacity efficiency, by adopting a failure-independent protection reaction. Once this chain of thought is followed it leads to the realization that the key to failure independent path-oriented *p*-cycles is just to enforce an a priori disjointness requirement on the end-to-end paths that share any *p*-cycle *structure*, just as SBPP enforces this on any primaries that share protection *channels*. Thus the key principle is:

> *Let the cycles act as p-cycles for end-to-end paths between nodes on the cycle, but only allow each cycle to provide protection relationships to a group of paths whose routes are all mutually disjoint.*

To illustrate, let us return to the network backdrop from Figure 2-14(a) and show how a certain set of working routes can be arranged to share a single FIPP *p*-cycle for their end-to-end protection, without any dependency on failure location. Figure 2-14(b,c,d) showed how a certain set of four primary paths which are disjoint from each other can share spare channels on several spans to help form any of their backup paths. Figure 8-1(a) shows an augmented set of end-node pairs (seven in total for the example) and routes between them, which includes the same subset of four demands from Figure 1, all of which can share the same FIPP *p*-cycle on their end-nodes.

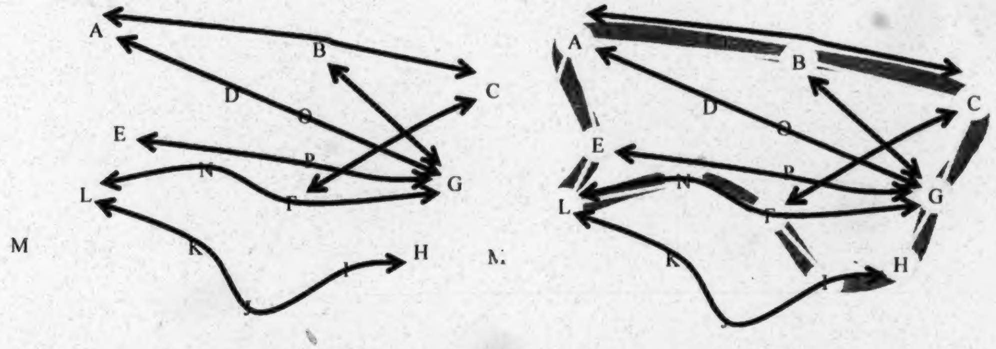


Figure 8-1. Example of a FIPP *p*-cycle protecting a set of seven mutually disjoint primaries. The whole pre-connected protection *structure* is shared intact. (Adapted from [KoGrJLT])

The important property that allows these seven working routes to be protected by one *p*-cycle in common is that they are all mutually disjoint. This is what leads to failure-independent end-to-end path protection via fully pre-connected structures of spare capacity. In a sense this draws directly from the key property underlying SBPP that in order to share spare *channels* on a span, all participating primaries must be mutually disjoint. Only what is different, and significant, is that by defining a *p*-cycle with respect to a group of mutually disjoint primaries, those primary paths are all enabled to share a fully pre-connected protection *structure*, not individual spare channels which still have to be cross-connected to form backup paths. Note also in the example that some of the routes in the group of disjoint routes selected fully straddle the respective FIPP *p*-cycle. These are the routes between end-nodes B-G, A-G, E-G and C-F. On these routes it will be possible to protect two working paths per unit of capacity on the *p*-cycle. One route, A-C is fully on-cycle and routes between end-nodes L-G and L-H are partially on-cycle and partially straddling. The capacity and switching implications of the latter two types of relationship between *p*-cycle and its protected routes are discussed further below. At this point, however, note from the description so far, including Figure 8-1, that all of the following properties arise in conjunction:

i) No cross-connections will be needed in real time to form the protection paths themselves. (Cross-connection for traffic substitution to break into and use the protection paths that the cycle provides, at the two end-nodes, is still required, however, as it is for any scheme not involving dedicated duplication such as 1+1 APS). This means that the utmost in speed is possible (again, 1+1 APS aside) and certainty that the optical path works when needed. As with any pre-connected protection structure, only two nodes (here the path end-nodes) need to act in real-time, to switch the affected traffic into and out of the backup path.

ii) Protection switching is entirely failure independent, end-node controlled, reacts to either span or node failure along the path, and only a single advance switching action is preprogrammed at each end-node of the path.

iii) Routes that straddle the cycle can each bear two working paths for each unit of spare capacity from which the *p*-cycle is formed.

iv) Protection of paths that transit a failed node is obtained if the group of working routes is required to be fully link and node disjoint. Otherwise node disjointness can be relaxed to link disjointness. (End-nodes common to both the working route and the protecting FIPP *p*-cycle is of course the exception. But under any scheme of network survivability, traffic is lost when the ultimate source-sink node of a path is itself failed.) These properties are identical to SBPP. For SBPP to provide node protection (as well as link) all primaries that share spare capacity on their backup routes must be fully disjoint, but otherwise can be only link disjoint.

v) The route taken by any signal in a failure-protected state can be fully known in advance. By the same token, it should be no harder than in regular *p*-cycle design to limit cycle size, to limit the length of any restoration path, or even specifically limit the length of any individual protection path.

vi) The protection structures in the network are a relatively small number of cycles, and are as easily visualized, changed and managed as a set of conventional cross-connect managed *p*-cycles for span-oriented protection.

It is immediately apparent that dynamic demand can be just as easily handled under the PWCE concept using path-protecting *p*-cycle structures as it is with conventional *p*-cycles.

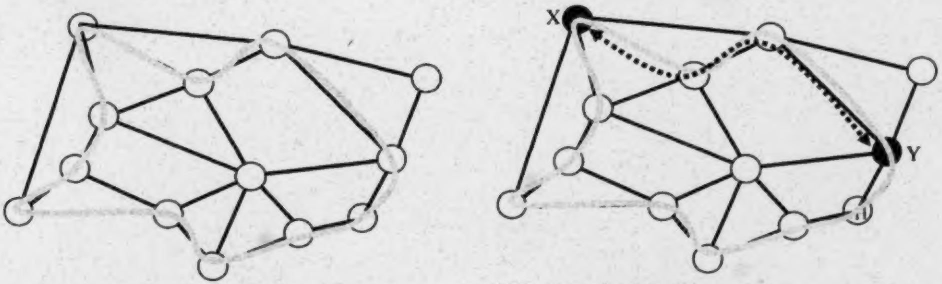
8.3.3. FIPP *p*-Cycle Operation

Basic operation is almost unchanged relative to ordinary *p*-cycles. To illustrate, Figure 8-2(a) gives a network context and an illustrative cycle for discussion. In Figure 8-2(b) the route is in a pure straddling relationship to its protecting *p*-cycle. In this case two working paths can be supported on the route (per unit of *p*-cycle capacity) and only the end-nodes of the demand have to do any switching. The assignment of each working path to the left or right-going protection path can be based on an odd-even assignment, or any other criteria. The pre-assigned switching direction is stored at the end-nodes in the same way that the path to *p*-cycle protection relationship itself is stored at each path end-node. Under optimized design, the bulk of the relationships between paths and chosen *p*-cycles will tend strongly to be of this type because it is twice as efficient as fully or partially on-cycle relationships and the solver is at liberty to chose cycles to relate to the routes to be served most often in this preferential way. This is the simplest case to appreciate the failure independence property because the pre-armed end-node switching action does not depend on the exact failure location or even the type of failure (i.e., span or node failure). By making sure that we consider only the protection of disjointly routed (i.e., "compatible") demands under any single *p*-cycle, we avoid the need for any failure information dissemination as no failure can possibly strike two demands protected by the same FIPP *p*-cycle. We also address the mutual capacity problem because no two members of a compatible demand set can fail simultaneously, there cannot be any contention for spare capacity on a *p*-cycle.

In Figure 8-2(c) the route is entirely on-cycle. Such cases are also simple: only one protection path is provided, and it is unambiguously determined as the other part of the cycle itself. Thus at the path end-nodes, the protecting cycle number is stored and only one path is associated with it, in the one unique direction that is usable.

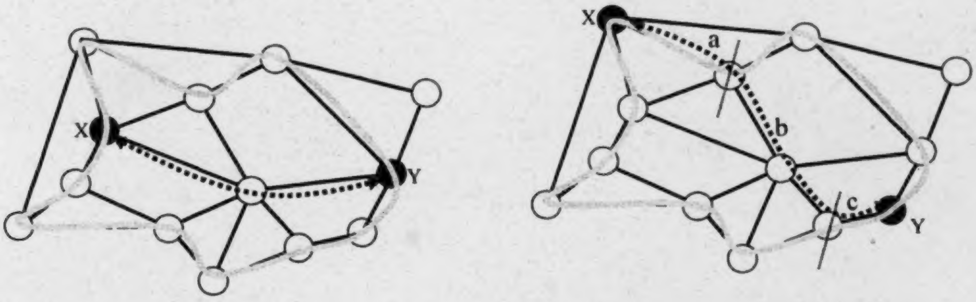
Figure 8-2(d) shows the most general case and one for which several approaches suggest themselves. It is the situation where the route bearing paths to be protected is partly on-cycle to the associated *p*-cycle. As mentioned, this case is limited to providing a single protection path per unit of *p*-cycle capacity, but we also need to address the switching logic. Let us say for instance that the "default" protection switching pre-plan for the assigned path on the route shown is to switch to the "lower" path segment of the cycle between nodes X and Y. Then if the path fails on segment *a* along its route, everything is fine because the lower protection path survives the failure. But if the path fails on segment *c* then the default switching direction is no longer viable because a failure on this segment also fails the related *p*-cycle itself in the "default" direction for that path. In this situation we need some way to know to change behavior and to restore the affected path now over the "upper" arc of the cycle. However, this is realized locally within the end-nodes (both X and Y) just by adding a rule that says "if the span in which the failed path arrives is the same as the span in the default direction on the *p*-cycle for that path, then use the other direction." In other words the coincidence of failure states (or appearance of Alarm Inhibit Signal (AIS)) on both a path and its respective protecting *p*-cycle as seen at the paths end-nodes is sufficient (and completely local) information to guide the choice of protection path. No special-case signaling is thus required as it might seem. of course in all cases, it is assumed and understood that the optical paths between nodes X and Y on the *p*-cycle are all optically feasible by design otherwise such cycles would not be admissible to the network design.

One further possibility not shown in Figure 8-2, but logically similar to Figure 8-2 (d) is where the route has one or more spans on-cycle to the FIPP *p*-cycle, but none of these are adjacent to the end nodes. In this case the pre-planning information still reflects that only one protection path can be offered, and its default direction. The need to change from the default direction can still be deduced locally at failure time, however, because either Loss of Light (in an optically transparent path) or AIS (in an opaque path) will propagate on the failed arm of the FIPP *p*-cycle to the respective end nodes controlling the switching. Thus, the failure is not limited to being in a span or path segment immediately adjacent to the end node to realize the *p*-cycle itself has been affected by the failure too.



(a) Network context and a FIPP *p*-cycle example

(c) Pure on-cycle relationship



(b) Pure straddling path relationship

(d) Partially straddling and partially on-cycle relationship

Figure 8-2. Different path to *p*-cycle topological relations for description of operation of FIPP *p*-cycles. (Adapted from [KoGrJLT])

Finally, in discussing the pre-planning information to be kept in nodes and their logical switching behavior, another overall strategy is worth noting. As so far described, there is a presumption of specific pre-planned configuration data being downloaded into each end-node, indicating which path is associated with which *p*-cycle, and in what direction. This is still not very complicated and may lead to the fastest possible switching times. It is also suitable for a completely transparent optical network where no channel-associated signaling between nodes is supported. But a more general and more self-managing approach is also possible, as follows; if we assume that path end-nodes can monitor channel-associated overhead bytes on both the working paths they terminate and on FIPP *p*-cycles passing through them. The first assumption is assured because the working paths are by definition terminated at their end-nodes. The second assumption may engender some extra cost at the cross-connect nodes to electrically monitor FIPP *p*-cycle channels. This may not, however, be much more expensive than the basic capability already required at any end-node, i.e., the ability to break into the optical *p*-cycle and both launch and receive a new electrical payload signal to/from the protection paths that the *p*-cycle is already providing. Assuming such channel-associated overhead signaling, every topologically distinct FIPP *p*-cycle is numbered when created. If multiple unit-capacity copies of the same cycle are built, they all bear the same unique FIPP *p*-cycle number, because this number is really just an identifier for a specific group of mutually disjoint routes in the network, which are all mutually disjoint and have end-nodes on that common topological cycle. This number and a list of end-node pairs with protection relationships on the cycle and a list of the spans in the cycle is applied to the overhead bytes and forever "march around in a circle" on each *p*-cycle itself, advertising its existence and the set of node-pairs (whose routes are all mutually disjoint) that are compatible with its use. In addition, each working path (while still in electrical form at its source) is labeled with the number of the FIPP *p*-cycle with which it is associated for protection. With this in place, even as network configuration changes, every node can easily learn and update the inventory of FIPP *p*-cycles currently available to it and can self-preplan the associations and default directions of each path for which it is an end-node to the available FIPP *p*-cycles. Any node can also know in advance of a failure if by some chance the current network configuration (i.e., set of working paths and configured *p*-cycles) did not actually provide for 100% restorability of any of its terminating paths.

But even more generally, another option is to collect the *p*-cycle number information off each terminating path, and observe the FIPP *p*-cycles that are passing through the node and then just wait until failures arise. Once a failure has manifested itself, the exact situation of failed working paths and surviving FIPP *p*-cycle segments is uniquely defined and known at all end-nodes by virtue of the alarm status in each port at each end-node, along with the pre-failure information collected. At this point a simple matching algorithm can be applied to assign each failed working path signal into an appropriate port where a surviving FIPP *p*-cycle segment is known to be present and offering a protection path to the desired peer end-node. By matching each failed working path to one unit of surviving capacity on a locally accessible FIPP *p*-cycle of the same topological number, the overall restoration action is inherently (although indirectly) cognizant of the mutual capacity coordination issue at the level of the network as a whole and also takes into account any non-default routing requirements arising from partially on-cycle relationships as in the case of Figure 8-2(d) above. In this approach of generalized node-local matching of failed working paths to available protection paths a multiple Quality of Protection priority scheme is easily derived. In addition, the scheme carries on working, making best use of available resources even when more than one failure arises because it is self-updating to the onset of additional working path failures and/or failure-related removal of FIPP *p*-cycles already in use from a first failure. The key to the generality is that each end-node continually knows the other end-nodes to which it has restoration needs, and knows the inventory of currently surviving protection paths to those other nodes, and the topological compatibility of each failed path with respect to shared use of the same FIPP *p*-cycle capacity.

8.3.4. FIPP *p*-Cycles and Non-Disjoint Routes

Because of the properties of *p*-cycles, inherited by FIPP *p*-cycles, strict mutual disjointness among protected working paths as illustrated in Figure 8-1(a) is not the only suitable condition. Each straddling working path could actually have up to two protection routes, because for such failures both sides of the *p*-cycle survive. If only one of the protection paths is actually used, then there may be an opportunity to consider protection (by the same FIPP *p*-cycle) of working paths that are not disjoint. The complete FIPP *p*-cycle principle can therefore be stated as:

> *Let the FIPP p-cycles act as conventional p-cycles for end-to-end paths between nodes on the cycle, but only allow each cycle to provide protection relationships to a group of origin-destination node pairs whose working routes: a) as a set, are all mutually disjoint or b) have disjoint backup routes on the cycle being considered.*

This principle is illustrated in Figure 8-3. In Figure 8-3 we see two sample working routes A-B routed along A-D-O-B shown by the thick arrow tipped line, and L-G, routed along L-E-D-O-P-G as shown by the dashed line. They are both routed over span D-O. However, if there is only one unit of capacity protected on each working route, and the end-nodes are pre-programmed to use the backup paths along the routes L-K-F-G for working path L-G and A-B for working path A-B, then there is no contention for spare capacity.

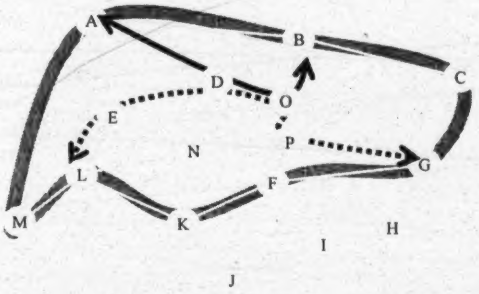


Figure 8-3. FIPP *p*-cycle protecting non-disjoint working paths.

However, there are a few reasons why we do not include this specific variation in either our design models or our results. First, it is easier to illustrate the FIPP *p*-cycle concept as pertaining to fully mutually disjoint routes (and hence use only the first principle) in the protected group of paths under any one FIPP *p*-cycle. A set of disjoint routes protected by a single cyclic structure is far more elegant than a complex set of routes that are protected on disjoint segments of the cycle. Incorporating this new principle in FIPP *p*-cycle design also proves to be very difficult as it significantly increases the complexity of the ILP model, which is already fairly complex. Recalling that *p*-cycles gain a two-for-one benefit from protecting capacity on straddling spans, as long as each protected route that fully straddles the cycle is loaded with twice the capacity of the corresponding cycle itself, this distinction is not expected to carry much of a capacity penalty[xxi]. From the results (presented later) that use only mutual disjointness as the defining condition, FIPP *p*-cycles are already close to corresponding path restorable solutions in capacity efficiency. Hence, for the models in this thesis, we only consider mutual disjointness as the defining characteristic of a FIPP *p*-cycle protected route set. Incorporating non-disjointness has been left to future work.

8.4. FIPP *p*-Cycle Network Design

Let us now ask how we would correspondingly design networks that operate based on this type of protection structures. To do so let us first define the concept of a group of disjoint routes to represent any set of working routes that are all mutually disjoint. The mutual disjointness applies to spans if the design aim is to protect only against span failures, otherwise the disjointness applies to both nodes and spans of the routes in the group. Any set of primary paths in SBPP that share at least one spare channel amongst their backup routes could be said to form such a group of disjoint routes. The significance is that any protection resource shared by paths routed over the members of a disjoint group of routes will never be in conflict with each other under any single failure scenario. In this framework, we can restate SBPP as a scheme for defining disjoint route sets for the sharing of individual spare channels and FIPP *p*-cycles becomes a scheme for defining disjoint route sets for the sharing of entire pre-connected protection structures. Given this orientation to viewing demands in groups based on their routing leads to the following basic ideas through which to approach the FIPP *p*-cycles network design problem.

One principle can be stated as: "Given a cycle considered as a candidate FIPP *p*-cycle, identify a subset of routes between end-nodes that are on the same cycle which never contend at the same time for restoration by the associated *p*-cycle (i.e., which form a disjoint route set on the end-nodes of the candidate *p*-cycle)."

A second principle with the same aim is: "Identify groups of routes over the graph which are all mutually disjoint. Then define a path-protecting *p*-cycle by routing a cycle through the collected set of end-nodes of these routes. Allow that *p*-cycle to be capacitated so as to protect all the working paths that the network's demand matrix requires to be routed over those routes."

We now describe design methods incorporating both these principles.

8.4.1. Spare Capacity Placement Design Model (FIPP-SCP) (First Principle)

Let us first define the term "*Disjoint Route Set*" (DRS) to represent any group of demands (routed along the shortest paths so the word demand and working route are used interchangeably in this subsection) that share no commonality in their working routing. Our current model approach, based on the first principle in 8.4 attempts to find the optimal DRSs that can be protected by a candidate *p*-cycle. The objective is to then find the optimal set of DRSs and *p*-cycles that lead to an overall minimum cost design. We first use standard methods to enumerate all the cycles of the graph. Eligible cycles are denoted *p* and *(x,y)* denotes the end nodes of any demand flow. We now define the following model:

Sets

S	Set of spans, indexed by *i* (failed) or *j* (surviving).
D	Set of demand relations, indexed by *r*.
P	Set of eligible cycles, indexed by *p*.

Parameters

Δ	A large positive constant (100000).
∇	A small positive constant (0.0001).
c_j	Cost of span *j*. (Can include all equipment costs and is proportional to length)

d_r	Number of demand units for relation *r*.
x_r^p	Equal to 1 if the demand relation *r* is on-cycle *p*, 2 if the demand relation *r* is completely straddling cycle *p*; 0 otherwise.
π_j^p	Equal to 1 if cycle *p* crosses span *j*, 0 otherwise.
$\partial_{m,n}$	Equal to 1 if demand relations *m* and *n* are rivals. This means *m* and *n* are not disjointly routed.

Variables

s_j	Spare capacity placed on span *j*.
n^p	Number of unit-capacity copies of cycle *p* in the solution.
n_r^p	Number of copies of cycle *p* used to protect demands on relation *r*.
γ_r^p	Equal to 1 if cycle *p* does protect demand relation *r*, 0 otherwise.

Objective

FIPP-SCP: Minimize: $\sum_{\forall j \in S} c_j \cdot s_j$ (48)

(Minimize total cost of spare capacity placed.)

Constraints

$$\sum_{\forall p \in P} x_r^p \cdot n_r^p \geq d_r \quad \forall r \in D \tag{49}$$

(The entire demand quantity for relation *r* must be protected)

$$n^p \geq n_r^p \quad \forall r \in D \tag{50}$$

(Place the maximum number of copies of cycle *p* required for any single demand.)

$$s_j \geq \sum_{\forall p \in P} n^p \cdot \pi_j^p \qquad \forall j \in S \tag{51}$$

(Place enough spare capacity to form all the *p*-cycles.)

$$\gamma_r^p \geq \nabla \cdot n_r^p \qquad \forall r \in D, \forall p \in P \tag{52}$$

(γ_r^p is 1 if n_r^p is greater than 0.)

$$\gamma_r^p \leq \Delta \cdot n_r^p \qquad \forall r \in D, \forall p \in P \tag{53}$$

(γ_r^p is 0 if n_r^p is 0.)

$$\partial_{m,n} + \gamma_m^p + \gamma_n^p \leq 2 \qquad \forall (m,n) \in D^2 \mid m \neq n; \forall p \in P \tag{54}$$

(Don't allocate the same cycle to protect two rival demands.)

Constraint (49) ensures that all demand for a particular O-D pair *r* is protected using *p*-cycles. Constraint (50) ensures that only the maximum number of instances of *p*-cycle *p* required for any single demand relation *r* is provisioned. Constraint (51) ensures that sufficient spare capacity exists to form all the *p*-cycles selected by the design. Constraints (52) and (53) define the binary variable γ_r^p which simply defines, at run-time, whether *p*-cycle *p* is indeed used to protect demand *r*. Constraint (54) is the key new FIPP specification that ensures that any individual *p*-cycle only protects a set of mutually disjoint working routes. If working routes for demands *m* and *n* are not disjoint (i.e $\partial_{m,n}=1$) then only one or none of γ_m^p or γ_n^p can be 1 at the same time- and consequently *p*-cycle *p* can only be used to protect one of *m* or *n*.

The above model is a mixed ILP model. One relaxation which we use is to drop the integrality requirement on the n_r^p variable. n_r^p is the number of unit working flows that the cycle *p* protects for demand pair *r*. In cases of a straddling node pair with an odd number of demand units, there may result an under-utilized *p*-cycle even as part of an optimal design. In this case the relaxed model would show fractional values for n_r^p but real fractional values would never arise in these circumstances because the working flows are themselves integral. In addition, since n^p, is kept as an integer it doesn't matter to the solution quality if n_r^p is fractional. This is just another instance of flow variable relaxation under integer capacity that is a useful and well-recognized technique in network design, which does not affect the ultimate solution quality (but reducing the number of integer variables speeds up the solution).

8.4.2. Joint Working and Spare Capacity Placement Design Model (FIPP-JCP)

In the FIPP-SCP model, demands were routed along the single shortest path and the solution was a grouping of FIPP *p*-cycles and DRSs that resulted in a minimum capacity design. However, in many previous works both on *p*-cycles and mesh, it has been found that allowing the solver to jointly select the working routing with the protection paths unlocks significant savings in total capacity requirements. FIPP *p*-cycles may also benefit from joint design and in this section we present a brief discussion about joint optimization of networks and list our first Joint Capacity Placement (JCP) model for FIPP *p*-cycle networks. In all, in this Chapter, we present two joint design models for FIPP *p*-cycle design.

8.4.2.1 Joint Optimization of Networks

Let us now review what it means to "jointly" optimize networks and review some relevant literature on jointness and its impact on network design. This creates the starting point for developing a joint design model for FIPP *p*-cycles.

In the literature on survivable mesh network design, the spare capacity optimization problem is often based on an Integer Linear Programming (ILP) model. The general aim is to determine the amount and distribution of *spare* capacity so that working capacity on each span can survive all single failures using minimum redundant resources. The working capacity values are given as inputs to the problem and arise from some routing policy such as shortest path routing or least loaded routing. In the joint approach the working and spare channels are simultaneously placed with the objective of reducing *total* capacity cost. Such Joint Capacity Placement (JCP) is typically more efficient than SCP because of the latitude to choose working routes other than strictly shortest routes or other fixed a priori policies, so as to maximize the overall effectiveness of spare capacity in protecting working capacity. Joint optimization of working and spare capacity has been found to yield reductions of as much as 28% in total capacity in mesh networks [IrMaTON98][XiMa99][DoGrDRCN01](relative to the corresponding non-joint designs). Similarly joint optimization in *p*-cycle networks achieved an improvement of about 25% [DoGrLEOS02].

A common notion is that the improved efficiency is attributable to the solver's ability to deviate the working path far from the shortest route. An insight from [IrMaTON98] and explained in [Grov03], however, is that (except for joint optimization with a strong economy of scale and modular capacity) working routes actually chosen in joint design are typically only a few percent longer than the shortest route in each case. [Grov03] (pp. 314-316) explains that jointness typically works more through load-leveling effects amongst almost shortest equal routes, and for good reason is very unlikely to find some significantly longer routes that are somehow highly synergistic with the spare capacity placement. The issue is that if a survivable network is fairly efficient even under SCP, say ~50% redundant for a simple illustration, then a working route choice that is x% longer than the shortest route for some demand, much lead to a spare capacity cost reduction of ~ 2x%. In other words, any increase in cost of routing a demand must be offset by a factor of ~two or more in spare capacity cost improvements in a 50% redundant network. Rather, joint optimization works more through exploitation of working capacity balance effects over multiple almost equal cost routes. We hypothesize that jointly optimized FIPP *p*-cycles network design must fundamentally exhibit the same behaviour and this helps explain our later results, and guide our strategy for deciding how many eligible working routes to represent in the problem. A related hypothesis that we make here is that there should be a threshold effect with respect to jointness where providing excess eligible working routes to the solver increases complexity but contributes no further improvement in solution quality. In the next section we develop the first of the two joint FIPP *p*-cycle network design models in this thesis.

8.4.2.2 FIPP-JCP Model

The FIPP-SCP model described previously considers shortest path[xxii] working routing and then chooses the *p*-cycles to protect each working capacity. But, as we saw in the previous section, the joint choice of working routes may improve the capacity efficiency.

This model is based on insights from prior *p*-cycle JCP models proposed in [GrDoLEOS02] and listed in [KoGrNFOEC03]. The solver now has the freedom to re-route the working demands to fully load up each individual *p*-cycle. For the joint model we define the following new sets, parameters and variables to the basic FIPP-SCP model in the previous section:

Sets

Q^r	Set of eligible working routes available for working paths of relation *r*.

Parameters

$\zeta_j^{r,q}$	Takes the value of 1 if the q^{th} working route for the demand pair *r* goes through span *j*.

Variables

w_j	Number of units of working capacity on span *j*.
γ_r^p	Equal to 1 if cycle *p* does protect demand relation *r*, 0 otherwise.
$g^{r,q}$	The number of working capacity units on the q^{th} route to satisfy the demand between node pair *r*.
χ_j^r	A temporary indicator variable that a route assigned to working route of demand relation *r* does lie over span *j*.
$\partial_{m,n}$	Equal to 1 if demand relations *m* and *n* are rivals. This means *m* and *n* are not disjointly routed.

Objective Function

FIPP-JCP: Minimize: $\sum_{\forall j \in S} c_j \cdot (s_j + w_j)$ (55)

(Minimize total cost of capacity placed.)

Constraints

$$\sum_{\forall p \in \boldsymbol{P}} x_r^p \cdot n_r^p \geq d_r \quad \forall r \in \boldsymbol{D} \tag{56}$$

(The entire demand quantity for relation *r* must be protected)

$$n^p \geq n_r^p \quad \forall r \in \boldsymbol{D} \tag{57}$$

(Place the maximum number of copies of cycle *p* required for any single demand.)

$$s_j \geq \sum_{\forall p \in \boldsymbol{P}} n^p \cdot \pi_j^p \quad \forall j \in \boldsymbol{S} \tag{58}$$

(Place enough spare capacity to form all the *p*-cycles.)

$$\gamma_r^p \geq \nabla \cdot n_r^p \qquad \forall r \in \boldsymbol{D}, \forall p \in \boldsymbol{P} \tag{59}$$

(γ_r^p is 1 if n_r^p is greater than 0.)

$$\gamma_r^p \leq \Delta \cdot n_r^p \qquad \forall r \in \boldsymbol{D}, \forall p \in \boldsymbol{P} \tag{60}$$

(γ_r^p is 0 if n_r^p is 0.)

$$\partial_{m,n} + \gamma_m^p + \gamma_n^p \leq 2 \qquad \forall (m,n) \in \boldsymbol{D}^2 \mid m \neq n; \forall p \in \boldsymbol{P} \tag{61}$$

(Ensure that two rival demands *m* and *n* are not protected by the same cycle *p*).

$$\sum_{q \in Q^r} g^{r,q} \geq d^r \qquad \forall r \in D \tag{62}$$

(sufficient working routing is considered)

$$\sum_{r \in D} \sum_{q \in Q^r} \zeta_j^{r,q} \cdot g^{r,q} \leq w_j \qquad \forall j \in S \tag{63}$$

(sufficient working capacity to support working routing)

$$\chi_j^r \geq \nabla \cdot \zeta_j^{r,q} \cdot g^{r,q} \qquad \forall r \in D \; \forall q \in \boldsymbol{Q^r} \quad \forall j \in S \tag{64}$$

(variable set to 1 if the actual working route of demand relation *r* is routed over span *j*)

$$\chi_j^r \leq \Delta \cdot \zeta_j^{r,q} \cdot g^{r,q} \qquad \forall r \in D \; \forall q \in \boldsymbol{Q^r} \quad \forall j \in S \tag{65}$$

(indicator variable set to 0 if no actual working route of demand relation *r* is routed over span *j*)

$$\partial_{m,n} \geq \nabla \cdot (\chi_j^m + \chi_j^n - 1) \qquad \forall m,n \in \boldsymbol{D^2} \,|\, \boldsymbol{m \neq n} \qquad \forall j \in S \tag{66}$$

(Set $\partial_{m,n}$ =1 if demands *m* and *n* have a common span in their working route.)

$$\partial_{m,n} \leq \Delta \cdot (\chi_j^m + \chi_j^n - 1) \qquad \forall m,n \in \boldsymbol{D^2} \,|\, \boldsymbol{m \neq n} \qquad \forall j \in S \tag{67}$$

(Set $\partial_{m,n}$ =0 if demands *m* and *n* don't have a common span in their working route.)

The new objective function in (55) ensures that the solver minimizes the overall total cost of capacity by considering simultaneous optimization of the working routing along with FIPP cycle placement. We can see in constraints (66) and (67) that $\partial_{m,n}$ is now a variable (it was a pre-computed parameter in the prior FIPP-SCP design constraint (54)). Constraints (66) and (67) together force $\partial_{m,n}$ to be *1* if demands *m* and *n* share any span *j* on *any* of their actual working routes *x* (for demand *m*) and *y* (for demand *n*). Constraint (61) then forces the cycle selection such that only demands that don't share any common span in their working routing are protected by the same cycle. The objective function requires lowering of spare and working capacity – and the solver automatically chooses the working routing (not necessarily shortest path routing) that yields a globally optimal set of FIPP *p*-cycles that would result in lower spare capacity. Constraints (62) to (65) are the joint constraints where the working demand is routed. Constraint (62) ensures that the demand bundle *r* is fully routed and constraint (63) ensures that sufficient working capacity is placed on the corresponding spans on which demand *r* is routed. Constraints (64) and (65) together force temporary variable χ_j^r to be 1 if the working route of demand relation *r* is over span *j*. Constraints (66) and (67) then ensure that if $\chi_j^m = 1$ and $\chi_j^n = 1$ for any span *j* then $\partial_{m,n}$ is set to 1 and the effect of this would be felt back in constraint (61) and then in (59) and (60) where the solver would not be allowed to provision any copies of a single cycle *p* to protect demands *m* or *n*.

Just by looking at the number of additional constraints (in comparison to the previous FIPP-SCP model), it is easy to see how FIPP-JCP is much more complex than FIPP-SCP. In fact, in its current form, it may be computationally very difficult solve for even the smallest of networks because of the sheer number of constraints. For example, the number of copies of constraints (66) and (67) are *o*((#eligible routes)^2 *(#demands)^2). This is in addition to the *o*((#demands)^2 *(#eligible cycles)) copies of constraints (61). of course the limiting case of there being only 1 eligible route per working path (single shortest path) is the SCP design discussed in the previous section. An avenue for further optimization (and future enhancement) of the JCP models is suggested by constraints (66) and (67) in the way the $\partial_{m,n}$ variables are formulated. The summation over all possible eligible working routes in constraint (62) allows the solver to split the demand bundle across multiple working routes. At the same time constraints (66) and (67) say that even if there is a conflict along one of the working routes of a demand pair *(m,n)*, then both demands cannot be protected by the same *p*-cycle. This is a subtle but important difference created by demand splitting in joint optimization. Thus, instead of a demand-bundle-wise view, a per-route constraint must be added. A subset of working routes of the demand bundles *m* and *n* may be actually disjointly routed and are thus protectable by the same FIPP *p*-cycle. This means that instead of $\partial_{m,n}$ we could have a $\partial_{m,n}^{r,s}$ variable which is set to 1 if working route *r* of demand *m* shares some part of its routing with route *s* of the demand *n*. We now only need to ensure that working routes *r* and *s* are not protected using the same *p*-cycle. However, to an already complex problem, this new consideration would add an additional set of (#demand^2*#eligible route^2) constraints. A possible workaround is of course to convert all demand bundles into differently enumerated unit sized demand-bundle copies. Another possible variation is to convert the n_r^p variable to be binary and enumerate multiple (more copies than the number of demands in the largest demand bundle) copies of each cycle in the input eligible cycle set numbered separately. Constraints (59) to (61) currently enforce that if one instance of *p*-cycle *p* is protecting a working route of demand relation *m* then no other copy of the same *p*-cycle *p* can protect a working route of another demand *m* if $\partial_{m,n} = 1$. Again, adding binary variables will increase ILP solution complexity and adding more eligible cycles will then further add to the number of constraints. From quick experiments, the solver produced no feasible solutions for this particular model.

We have therefore not pursued this approach to the FIPP-JCP model further, but use the second principle from 8.4 to develop a simpler joint ILP model called the FIPP-JCP-DRS in 8.4.3. We include the FIPP-JCP model only for theoretical completeness and only as a tool for further scientific precision if actually implemented in its present form. But the joint design suggests several uses – such as the possible use as a heuristic to refine an SCP solution by altering working routes to eliminate *p*-cycles, an ILP formulation to first obtain the set of maximally disjoint compatible routes and then use a reduced form of the joint model to obtain a semi-optimal result.

8.4.3. Disjoint Route Set Approach To FIPP *p*-Cycle Network Design (Second Principle)

In contrast to the approach in 8.4.1, we now use the second principle and form DRS' as candidates and select corresponding FIPP *p*-cycles that unify their end-nodes. But as before, we consider using the shortest working route for each demand relation in the network, identifying groups of mutually disjoint routes and forming candidate DRSs from them. Enumerating all possible DRSs as design candidates would be ideal, but a subset would be suitable in practice. We then ensure that each working route belongs to at least one candidate DRS, but preferably many. Finally, we build efficient FIPP *p*-cycles by routing a cycle through the collected set of end-nodes of the member demands of a DRS and capacitate all cycles so as to protect all the working paths that the network's demand matrix requires.

We now propose an ILP design model based on this "DRS enumeration" approach to the problem. This model, when solved with the set of all possible DRSs as input to the solver, should strictly produce the same results as the FIPP-SCP design.

The ILP model uses the following additional notation:

Sets

S is the set of all spans in the network, and is indexed by i for a failed span, and j for a surviving span.

C is the set of all eligible DRSs, and is indexed by c.

D is the set of demand relations, indexed by r.

$C_r \in C$ is the set of all DRSs that contain the working route of demand relation r, and is indexed by c.

P is the set of eligible cycles, indexed by p.

$P_c \in P$ is the set of eligible cycles that can protect DRS c, and is indexed by p.

Parameters

c_j is the cost of span j.

d_r is the number of unit demands in the bundle of demand relation r.

$x_r^p \in \{0,1,2\}$ encodes the relationship between a demand's working route and eligible p-cycles. $x_r^p = 1$ if demand r's end-nodes are on cycle p and its working route passes over at least one span on cycle p, $x_r^p = 2$ if demand r's end-nodes are on cycle p and its working route straddles cycle p; and $x_r^p = 0$ otherwise.

$\pi_j^p \in \{0,1\}$ encodes a cycle's constituent spans. $\pi_j^p = 1$ if cycle p crosses span j, and $\pi_j^p = 0$ otherwise.

Variables

$s_j \geq 0$ is the spare capacity placed on span j.

$n_c^p \geq 0$ is the number of unit-capacity copies of cycle p used as a FIPP p-cycle to protect DRS c.

$n^p \geq 0$ is the total number of unit-capacity copies of cycle p used as a FIPP p-cycle for all DRSs.

The ILP formulation of the DRS-based FIPP *p*-cycle network design model (FIPP-DRS) is as follows:

$$\textbf{FIPP-SCP-DRS}\text{: Minimize } \sum_{\forall j \in \boldsymbol{S}} c_j \cdot s_j \tag{68}$$

$$\text{Subject to} \qquad \sum_{\forall c \in \boldsymbol{C},} \sum_{\forall p \in \boldsymbol{P}_c} x_r^p \cdot n_c^p \geq d_r \qquad \forall r \in \boldsymbol{D} \tag{69}$$

$$n^p = \sum_{\forall c \in \boldsymbol{C}} n_c^p \qquad \forall p \in \boldsymbol{P} \tag{70}$$

$$s_j \geq \sum_{\forall p \in \boldsymbol{P}} n^p \cdot \pi_j^p \quad \forall j \in \boldsymbol{S} \tag{71}$$

The objective function minimizes the total cost of placing spare capacity in the network. For simplicity, we can equate c_j with the length of the span as drawn in the network graph, and in general, c_j represents costs of fiber, rights-of-way, amplifiers, etc. The constraints in (69) ensure that for each demand relation, r, a sufficient number of FIPP *p*-cycles are assigned to protect all selected DRSs of which the working route of demand r is a member. Equation (70) calculates the total number of copies of FIPP *p*-cycle p as equivalent to the sum of the numbers required for each selected DRS individually. Finally, (71) ensures that sufficient spare capacity exists to build the *p*-cycles selected by the design.

8.4.3.1 Joint, Modular Design of FIPP *p*-Cycle Networks Using The DRS Approach

In this section, we extend the FIPP-SCP-DRS model to a joint, modular FIPP-JCP-DRS model. Modularity and Economy of Scale are described in [LEOS02].

Sets

$\boldsymbol{S}$ is the set of all spans in the network, and is indexed by *i* for a failed span, and *j* for a surviving span.

$\boldsymbol{C}$ is the set of all eligible DRSs, and is indexed by *c*.

$\boldsymbol{D}$ is the set of demand relations, indexed by *r*.

$\boldsymbol{M}$ is the set of modules, indexed by *m*.

$\boldsymbol{C}_q \in \boldsymbol{C}$ is the set of all DRSs that contain working route *q*, and is indexed by *c*.

$\boldsymbol{P}$ is the set of eligible cycles, indexed by *p*.

$\boldsymbol{P}_c \in \boldsymbol{P}$ is the set of eligible cycles that can protect DRS *c*, and is indexed by *p*.

$\boldsymbol{Q}^r$ is the set of eligible working routes available for working paths of demand relation *r* and is indexed by *q*.

Parameters

d_r is the number of unit demands in the bundle of demand relation *r*.

$x_q^p \in \{0,1,2\}$ encodes the relationship between the q^{th} working route and eligible *p*-cycle *p*. $x_q^p = 1$ if working route *q*'s end-nodes are on cycle *p* and the route passes over at least one span on cycle *p*, $x_q^p = 2$ if working route *q*'s end-nodes are on cycle *p* and its working route straddles cycle *p*; and $x_q^p = 0$ otherwise.

$\pi_j^p \in \{0,1\}$ encodes a cycle's constituent spans. $\pi_j^p = 1$ if cycle *p* crosses span *j*, and $\pi_j^p = 0$ otherwise.

$\zeta_j^{r,q}$ Takes the value of 1 if the q^{th} working route for the demand pair *r* goes through span *j*.

$\zeta_j^{r,q}$ Takes the value of 0 if the q^{th} working route does not belong to any eligible DRS. This is necessary because not all routes may be chosen as part of at least one DRS.

Z^m Capacity of m^{th} module type.

C^m is the cost of a single module of type *m*.

c_j is the cost of span *j*.

Decision Variables

$s_j \geq 0$ is the spare capacity placed on span *j*.

$w_j \geq 0$ is the working capacity placed on span *j*.

$n_c^{p,q} \geq 0$ is the number of unit-capacity copies of cycle p used as a FIPP *p*-cycle to protect working route *q* as part of DRS *c*.

$n_c^p \geq 0$ is the number of unit-capacity copies of cycle *p* used as a FIPP *p*-cycle to protect DRS *c*.

$n^p \geq 0$ is the total number of unit-capacity copies of cycle *p* used as a FIPP *p*-cycle for all DRSs.

$g^{r,q}$ The number of working capacity units on the q^{th} route to satisfy the demand between node pair *r*.

t_j^m Number of modules of type *m* placed on span *j*.

FIPP-JCP-DRS: Minimize $\sum_{m \in M} \sum_{\forall j \in S} (c_j \cdot C^m \cdot t_j^m)$ (72)

Subject to:

$$\sum_{q \in Q^r} g^{r,q} \cdot \Omega^q \geq d^r \qquad \forall r \in D \qquad (73)$$

$$\sum_{r\in \boldsymbol{D}}\sum_{q\in \boldsymbol{Q}^r} \zeta_j^{r,q} \cdot g^{r,q} \le w_j \quad \forall j \in S \tag{74}$$

$$\sum_{\forall c\in \boldsymbol{C}_q}\sum_{\forall p\in \boldsymbol{P}_c} x_q^p \cdot n_c^{p,q} \ge g^{r,q} \quad \begin{matrix} \forall r \in \boldsymbol{D} \\ \forall q \in \boldsymbol{Q}^r \end{matrix} \tag{75}$$

$$n_c^p \ge n_c^{p,q} \quad \begin{matrix} \forall r \in \boldsymbol{D} \\ \forall q \in \boldsymbol{Q}^r \\ \forall c \in \boldsymbol{C}_q \\ \forall p \in \boldsymbol{P}_c \end{matrix} \tag{76}$$

$$n^p = \sum_{\forall c\in \boldsymbol{C}} n_c^p \quad \forall p \in \boldsymbol{P} \tag{77}$$

$$s_j \ge \sum_{\forall p\in \boldsymbol{P}} n^p \cdot \pi_j^p \quad \forall j \in \boldsymbol{S} \tag{78}$$

$$w_j + s_j \le \sum_{m\in \boldsymbol{M}} t_j^m \cdot Z^m \quad \forall j \in S \tag{79}$$

The objective function in (72) minimizes the total modular cost of placing spare and working capacity in the network. For simplicity, we can equate c_j with the length of the span as drawn in the network graph, and in general, c_j represents costs of fiber, rights-of-way, amplifiers, etc. C^m represents the cost of module *m*. C^m is used to include economy of scale effects. Constraints (73) and (74) are the working routing constraints. Constraint (73) ensures that the entire demand bundle *r* is fully routed along one or more of the *q* working routes available for demand bundle *r*. Constraint (74) then ensures that sufficient working capacity is placed on the network spans to support all working routing over that span simultaneously. Constraint (75) then ensures that sufficient number of FIPP *p*-cycles are placed to protect all selected DRSs. Constraints (76) and (77) calculate the total number of copies of FIPP *p*-cycle *p* as equivalent to the sum of the numbers required for each selected DRS individually. There is no requirement that different DRSs be mutually disjoint. Thus a single cable cut may affect multiple working routes that belong to different DRSs. Finally, (78) ensures that sufficient spare capacity exists to build the *p*-cycles selected by the design. Constraint (79) incorporates modularity and economy of scale into the problem. As a special relaxation, we can relax $n_c^{p,q}$ and

n_c^p because the number of *p*-cycles that will actually be built is bounded by n^p which is integral. Thus $n_c^{p,q}$ and n_c^p are also implicitly integers.

8.5. Experimental Setup

In this section we describe the experimental setup for the various design experiments using the models described in section 8.4.

8.5.1. Experimental Setup For FIPP-SCP ILP Model

The FIPP-SCP design model (8.4.1) was coded in AMPL™ and solved using CPLEX™ on a dual Opteron Windows 2000 machine with 1 Gigabyte of RAM. The test network used is the COST239 European network from [COST239] reproduced in Figure 8-4. Figure 8-4 also indicates span numbers for reference in the later discussion later in 8.6. The cost of each span is assumed to be 1, so in this case the minimum cost design is the same as the minimum capacity design. This network has 11 nodes and 26 spans. A pre-processing program written in C++ initially produces a list of candidate cycles in the graph. There are somewhat over 3000 distinct simple cycles possible in the COST239 network and we choose the 1000 longest cycles as candidate *p*-cycles. Working capacity is routed via shortest paths and we then pre-calculate the x_r^p, π_j^p and $\partial_{m,n}$ parameters based on the cycle and working routing information. This information is provided as input to the AMPL-CPLEX solver. The demand matrix is varied for different test cases from a sparse 19 demands to the maximum possible 55 demands in the 11-node COST239 network. The 19 and 27-demand test cases have demands between randomly chosen node pairs. We will later examine in detail the 19-demands test case, and hence we list all the 19 demands, their end-nodes and details of their working routing Table 8. The 27 demand test case also has randomly generated demands from the total of 55 possible demands. For all test cases, each demand is exactly 2 units. SBPP benchmark designs are based on the optimal SBPP-SCP design model from [DoClONM03] with all routes of equal hop-length which are disjoint from the working route included as backup route candidates for each SBPP primary route. Ordinary span-protecting *p*-cycle SCP reference designs were also produced based on an optimal design model from

[KoGrNFOEC03] with a 1000 shortest cycles as input. The demand matrix and working routing was the same for all the different test cases. Overall the FIPP-SCP designs took the longest time to solve – which is as expected because of the higher complexity. The SBPP and *p*-cycle benchmark designs are solved to within 1% MIPGAP. The FIPP *p*-cycle designs were also set at 1% MIPGAP, but despite running for over 2 days and multiple trials and variable relaxations, the integer feasible solutions were between 30% and 60% away from the fully-relaxed lower bound. This means that good feasible designs were obtained by the AMPL-CPLEX optimization, but we do strictly know only that these designs are within 30% to 60% of optimal. They are, however, probably fairly close to optimal solutions because it is generally known that in problems with many 1/0 variables the fully relaxed LP lower bound is usually quite loose. In any case, sub-optimality only works against the claims we make for FIPP *p*-cycle efficiency in the comparative design results. At present, our concern is not about the run time, but the properties of this new network architecture itself, using ILP as a tool to simply to reveal its intrinsic efficiency compared to other schemes.

But even with this handicap, the FIPP *p*-cycles were remarkably capacity efficient. In the 19 and 55 demand test cases, FIPP *p*-cycles outperformed span *p*-cycles. In the 27 demand test case, the result was very close. The results for the SBPP design and the FIPP-SCP designs were also remarkably close together.

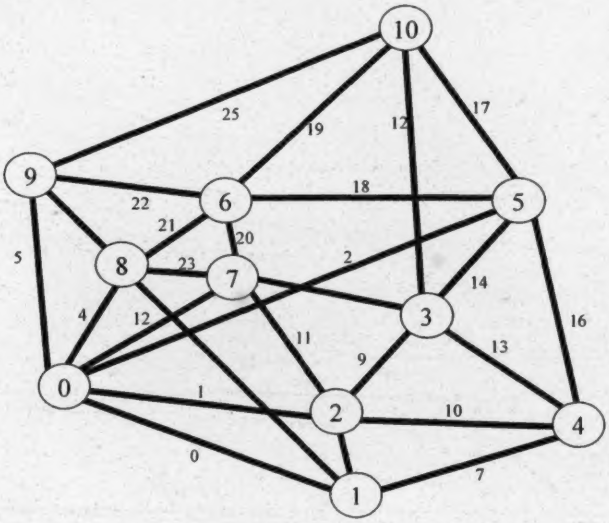


Figure 8-4. Cost 239 test network indicating span numbers. (Adapted from [BaCOST239])

Table 8. Demand matrix for test network. (only non-zero demands listed) (Adapted from [KoGrJLT])

Demand	End Nodes	Spans in working route
0	(0,1)	0
1	(0,4)	0,7
2	(0,7)	3
3	(0,10)	2,17
4	(1,4)	7
5	(1,7)	0,3
6	(1,10)	6,9,12
7	(2,5)	1,2
8	(2,8)	1,4
9	(3,4)	13
10	(3,7)	15
11	(3,10)	12
12	(4,7)	10,11
13	(4,10)	13,12
14	(5,8)	2,4
15	(6,7)	20
16	(6,10)	19
17	(7,10)	15,12
18	(9,10)	25

8.5.2. Experimental Setup For FIPP-SCP-DRS ILP Model

To obtain a strictly optimal FIPP *p*-cycle network design solution with the FIPP-SCP-DRS ILP model, the solver would strictly require all possible DRSs as input. This is of course combinatorially quite complex. Enumerating all DRSs would make the FIPP-SCP-DRS model just as difficult to solve as the FIPP-SCP design. Consider that a network with n nodes has $K = n\cdot(n-1)/2$ node pairs to exchange demand. There are potentially ${}^{K}C_{2}$ possible combinations of 2-route groups, ${}^{K}C_{3}$ possible 3-route groups, and so on, for a total of $\sum_{\forall i\in\{1\ldots n\}} {}^{K}C_{i}$ if we require node-disjointness, or $\sum_{\forall i\in\{1\ldots s\}} {}^{K}C_{i}$ if span-disjointness is sufficient (where s is the number of spans in the network). It is an intractable problem, of $O(n^{2}!)$, just to enumerate all possible combinations of routes from which to select the DRSs, and this is only if each demand relation considers a single eligible working route. We therefore use an algorithmic approach to partially enumerate a promising and practically-sized DRS set. The basic algorithm is in Table 9. The implementation we tested for the results below, ran the algorithm in Table 9 ten times, so that each working route, *r*, was included in at least ten individual DRS candidates. We also added a random termination function such that each time through the algorithm, there was an increasing probability that we would mark the current DRS as complete after each new route was added to it. The result is that some DRSs include many disjoint routes (as many as 20), while others include as few as two or three. We also add to the set of DRSs an additional "ultimate recourse" DRS associated with each node-pair individually. These DRSs each contain only the one shortest working route for each node pair individually. This allows that the model is able to select a FIPP *p*-cycle solely for the protection of a single highly-sized working route, if that is necessary for feasibility or optimality. Then for each DRS in this set of DRSs, we find the 10 shortest eligible cycles that join all end-nodes of the DRS (if ten are feasible). In doing this we are careful to allow any specific eligible cycle to protect other DRSs as feasible, not just the one for which it was enumerated.

Table 9. FIPP-DRS-SCP Generate DRS heuristic algorithm. (Adapted from [KoGrDRCN05])

```
GenerateDRSs() {
  Initialize DRSSet
  For each working route r{
    Unmark all working routes
    Mark working route r
    While DRS c not complete {
      Randomly select unmarked route, x
      Mark working route x
      If x is disjoint from c {
        Add working route x to DRS c
      }
      If no unmarked working routes remain {
        DRS c is complete
      }
    }
    Add DRS c to DRSSet
  }
  Return DRSSet
}
```

The ILP model was implemented in AMPL 9.0 and solved using the CPLEX 9.0 MIP solver on a dual-processor AMD Opteron 242 PC with 1 Gigabyte of RAM running Windows 2000. On this platform FIPP-DRS runtimes were in the order of seconds or a few minutes for most test cases, benchmark comparison SBPP runtimes were several minutes, and *p*-cycle runtimes were several seconds at most. The test case networks used are the 15-node family of related networks and their associated demand matrix (full mesh of demands with each node pair exchanging a uniform random demand from 1 to 10) from [DoPhD04], reproduced in Figure 8-5

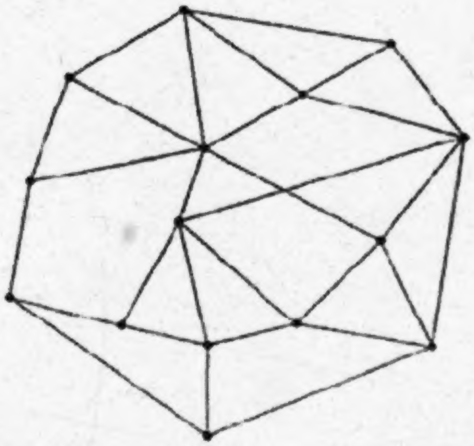


Figure 8-5. 15n30s Test Network (used for FIPP-JCP-DRS trials) (Adapted from [DoPhD04])

The network family is headed by a *master network* with 30 spans (for an average nodal degree of 4.0). Each other member of a family is obtained by successively applying individual pseudo-random span removals while keeping all nodal positions fixed. Any one network in the family is therefore identical to the next higher-degree network in the family except that one span has been removed. The same demand matrix is used for each individual member of the family. In all cases, working routing is via shortest paths, and all variables are strictly integer. Benchmark SBPP and *p*-cycle designs were produced using the same ILP models as described in [DoPhD04], and with the same design parameters: SBPP designs considered the 10 shortest backup routes for each demand relation, and the *p*-cycle designs had 1000 eligible cycles (the 1000 shortest that can possibly be drawn in the graph). FIPP-DRS and SBPP results are based on full CPLEX terminations with optimality gap settings of 0.02, or in other words, all solutions are solved to be within 2% of optimal given the specified inputs. *p*-Cycle results were obtained with optimality gap settings of 0.0001 (within 0.01% of optimal).

8.5.3. Experimental Setup For FIPP-JCP-DRS ILP Model

8.5.3.1 Modified Generate DRS Heuristic

For the FIPP-JCP-DRS model, we again only partially enumerate the set of DRSs for the same reasons as discussed in 8.5.2. The creation of candidate DRSs is based on a heuristic algorithm that is a modified version of that in Table 9. The modified GenerateDRS algorithm is in Table 10.

Table 10. Modified GenerateDRS algorithm for the FIPP-DRS-JCP model (Adapted from [KoBaPNET06])

```
ModifiedGenerateDRSs() {
Initialize DRSSet
Initialize EligibleRouteSet
  For each Demand r {
    Generate n shortest working routes between the nodes of  r and add them to
    the EligibleRouteSet
  }

  For each Demand r {
     For Number of DRSs per Demand needed {
         Add shortest route of Demand r to DRS c
         While DRS c not complete {
            Randomly select a working route  x from the EligibleRouteSet
                    If x is disjoint from c {
                           Add working route  x to DRS c
                    }
            RandomExit()
         }
        Add DRS c to DRSSet
     }
  Return DRSSet
 }
```

8.5.3.2 Standard Parameters

For each test case (unless specified otherwise), for each demand, we found the first five shortest eligible working routes by length. All such routes were grouped into a master set of eligible working routes. As discussed in the generateDRS algorithm in Table 10, the shortest working route for each demand was taken as the starting point for a sample DRS and additional routes (from the master eligible working route set) were chosen such that they are disjoint from all the routes already in the DRS. For each demand, the generateDRS Algorithm was repeated 10 times resulting in 10 eligible DRSs for that demand. For each DRS the five shortest cycles that pass through the collective set of end-nodes of the working routes in the DRS are enumerated. We also added, to the master set, a single 'bypass' DRS for each demand that contains only its single shortest route for the same reason as discussed in 8.5.2. This is done to eliminate forcer demands from the design problem. Forcer demands are defined as a demand that causes a net increase in design cost if the corresponding demand bundle is increased by one unit. Removing or eliminating a forcer by using dedicated protection techniques in the network may reduce total network cost. Forcer elimination is an even more important strategy to improve solution quality in heuristics. This bypass recourse allows the solver to place smaller dedicated *p*-cycles to protect large demand bundles and essentially remove them from the problem, if that results in a net decrease in network cost. Experiment specific variations are described in the Results section.

8.5.4. Test Networks and Computational Details

The ILP model was implemented in AMPL 9.0 and solved using the CPLEX 9.0 MIP solver on a quad-processor Sun V480 SPARC server with 16 GB of RAM. The test case network used is the COST239 network from Figure 8-4. The demand matrix comprised of 15 randomly chosen O-D pairs with a demand of 4 units between them. Span costs are approximated as the geographic distance between the nodes.

On the hardware given, none of the designs took longer than 2 hours to run to optimality. Each experiment was repeated five times to test heuristic robustness unless otherwise specified. In each graph of results below, individual points are plotted along with line through the average design cost and the lowest design cost for each run. All results are based on full CPLEX terminations with terminating MIPGAP of 0.01, or in other words, all solutions are solved to be within 1% of optimal.

In the nominal case, for each demand, we found the five shortest eligible working routes by length and grouped them into one eligible working route set (EligibleRouteSet). The shortest working route for each demand was taken as the starting point for a sample DRS and additional routes (from the eligible working route) were chosen such that they are disjoint from all the routes already in the DRS. We repeated this process 10 times for each demand and created 10 eligible DRSs for each demand. This also ensured that the shortest working route for the demand belonged to at least 10 DRSs in the final eligible DRS set. For each DRS we then found the five shortest cycles that pass through the collective set of end-nodes of the working routes in the DRS. To the set of eligible DRSs generated using the GenerateDRSs algorithm, we added a set of DRSs that contained the single shortest working route for each demand in the demand matrix. This was done to allow the solver to place smaller *p*-cycles to protect large demand bundles if that results in a net decrease in network cost. A random termination function was added to the GenerateDRSs algorithm. This randomly terminates the DRS formation loop, even if the DRS is not complete. This resulted in a selection of cycles of various sizes.

The ILP model was implemented in AMPL 9.0 and solved using the CPLEX 9.0 MIP solver on a quad-processor Sun V480 SPARC server with 16GB of RAM. The test case network used is the COST239 network in Figure 8-4. We considered 15 unique demand bundles between randomly selected node pairs and in all cases the demand bundle is exactly 4 units. Span costs are equal to the geographic distance between their end-nodes on the network graph. For the results below, we varied the number of routes per DRS, number of eligible cycles per DRS and the number of eligible working routes per demand one at a time and studied the effect on the network costs and the spare to working capacity redundancies. Each of the tests were repeated five times with the input set of DRSs generated randomly for each run. Exact data points are plotted individually and an average is drawn as a line in the graph for all the test cases. All variables are restricted to be integers. All results are based on full CPLEX terminations with optimality gap settings of 0.01, or in other words, all solutions are solved to be within 1% of optimal given the specified inputs. Runtimes were not recorded but they were in the order of seconds or minutes for most test cases, with none running over 2 hours.

8.6. Results for the FIPP-SCP Model

8.6.1. Preliminary Comments

Overall the FIPP designs took the longest time to solve, as compared to the benchmark SBPP and SCP designs and could not be solved to optimality for the larger networks. This is attributed to complexity arising primarily out of the γ_r^p 1/0 variables. In time these problems may be overcome with suitable branching strategies and added relaxations or bounds. often in light of the difficulty of ILP problems, people dismiss the use of ILP for network design altogether. At present, however, our concern is not primarily about run time, but in trying to appraise the intrinsic properties of this new network architecture itself, and for this ILP is an essential tool. In this regard the role of ILP is as a research tool to try to understand the intrinsic efficiency of the new scheme compared to other schemes. The philosophy is that if the intrinsic capabilities or potential of the new architecture as revealed by ILP studies are promising, it *then* makes sense to work on faster heuristic algorithmic design strategies and so on. So both ILP and design heuristics have their roles to play. But the problem with using heuristics for design of the corresponding test case networks for research comparison is that the results then depend on *both* intrinsic properties of the architectures being compared *and* the different sub-optimalities inherent in the various heuristics used. Even time-limited "best feasible" results from incomplete ILP runs can be valuable in research, however, as they usually still provide very high quality existence-proof type solutions, without even needing to develop a purpose-specific heuristic. For this purpose, we first attempt to compare FIPP *p*-cycles with other architectures like SBPP, span *p*-cycles and mesh using optimal solutions. In the later sections where we present results for the FIPP-JCP-DRS model, we will present a heuristic for FIPP *p*-cycle design.

In addition our philosophy is that differences in capacity efficiency of a few percent in are not really important in practice, if other important benefits such as full pre-connection can be achieved. It is more a matter of theoretical understanding of the relative schemes being compared. In practice our only real concern is to see if FIPP *p*-cycles are at least characteristically in the same range least as SBPP for efficiency.

Because if they are, it means their other advantages over SBPP can be exploited without practical penalty. Additionally, even if FIPP only performs better than span *p*-cycles and span restoration it is still of interest because now we have the *p*-cycle advantage of pre-connection *and* the ability to do end-to-end service protection.

Our initial tests are for the 7n12s test case from Figure 7-4(a) for which the solver terminated at a 3% MIPGAP for the Span Restoration (SR), Path Restoration (PR) and SBPP runs with complete problem information as above. For the larger networks, COST239 from [COST239] and illustrated in Figure 8-4 and 15n20s, (a 20 span derivative of the 15n30s network in Figure 8-5 from [DoPhD04]) the number of eligible routes and cycles was limited as described above. CPLEX terminations with a 1% MIPGAP were obtained for all the reference architectures but for these test cases the FIPP design results are only the "best feasible" solutions found after about two days of run time. These best feasible solutions remain associated with gaps of 30% to 60% above the fully-relaxed LP lower bound. This means that the solver itself can only assure us that these designs are within 30% to 60% of optimal. It needs to be kept in mind therefore, that in what follows for the two larger networks, feasible but sub-optimal FIPP *p*-cycle designs are being compared to other designs that are known to be within 1% of optimum.

8.6.2. Discussion

We start with a discussion of the results for the 7n12s in Table 11 using all possible eligible routes for the SR, PR and SBPP designs and all possible cycles as eligible cycles for the FIPP and span *p*-cycle designs and a 1% MIPGAP. On 7n12s we could only get within 3% of optimality for the test case where we used shortest path routing (first column of Table 11). Nonetheless, the FIPP design is within 7% of the corresponding SBPP-shortest path solution (32 versus 30).

Table 11. Total spare capacity results (channels required) for 7n12s network designs (Adapted from [KoGrJLT])

Protection Scheme	Shortest Path Routing	Adjusted Routing
FIPP	32(3% gap)	26
SBPP	30	26
SR	42	35
Span *p*-Cycles	42	35
PR	25	23

Table 11 also shows that PR also outperforms both FIPP and SBPP, and this is in line with expectations because as discussed, true path restoration is indeed the absolute lower bound on the capacity requirements for any single failure survivability technique. SR and span *p*-cycles perform equally well but worse than SBPP, PR and FIPP. This is also as expected from much prior research on these schemes. Recognizing, that these are non-joint designs, and because we have reason to suspect qualitatively that joint optimization may be relatively quite advantageous under the FIPP architecture, we allowed ourselves a further experiment on 7n12s where we manually deviated four of the working routes from their shortest routes to routes that would (by inspection) be equivalent in hop count, more amenable to efficient FIPP implementation. We then re-solved all the corresponding benchmarks for the new routing and the results are in the last column of Table 11. Here we found that the FIPP solution ran to a full termination under the 1% MIPGAP condition (in less than an hour) and is as good as the optimal SBPP design for the same set of demands and routes. Interestingly, all the other architectures also benefited, to varying degrees, from the adjusted working path routing. We consider this an experimental validation of the qualitative reasoning suggesting that FIPP and SBPP are at least categorically similar in the capacity efficiencies that can be achieved. By categorical similarity we mean, for example, in the sense that SR and span *p*-cycles are also categorically similar. The existence of an identical result, and another that is only ~3% different suggests that there is no argument that the schemes lie in theoretically different categories of capacity efficiency, such as can be stated with confidence about, for example, SR and PR, which are clearly in different basic categories of achievable efficiencies.

Results for the larger COST239 and 15n20s test networks are summarized in Table 12. In Table 12 the FIPP results are only the best feasible solutions obtainable in the time available. The bracketed value shows the remaining gap from optimality. In practice, where the remaining gap on a FIPP solution suggests the true optimum solution could be below the PR solution value, PR more correctly provides the real lower bound on the possible solution value for FIPP. The last row of Table 12 is for the special case of SBPP designs in which we allowed only routes equal in length to the single shortest backup path (disjoint from the working route) in the eligible route set. This represents a practical SBPP provisioning option (optimally designed nonetheless) wherein each working route takes the shortest path to the destination and its backup path is planned along *one of the next shortest routes* that is disjoint from the primary.

Table 12. Results (spare capacity channel counts) for Cost 239 and 15n20s networks (Adapted from [KoGrJLT])

	COST239-19	COST239-27	COST239-55	15n20s – 21
best feasible FIPP (gap)	33(30%)	44(47%)	67(66%)	173(58%)
SBPP(opt)	28	30	46	134
Span Restoration (opt)	29	33	62	174
Span p-cycles (opt)	39	47	76	174
Path Restoration(opt)	21	22	35	111
SBPP(SP Alternate-opt)	46	44	64	174

In COST 239, the best feasible FIPP designs are similar or better than SBPP when the latter uses only shortest alternate backup routes. In COST 239 with 19 demands, the best feasible FIPP design with a gap of 30% is within 18% of fully optimal SBPP. Beyond this, the large remaining gaps on the best-feasible FIPP *p*-cycle designs make it hard to say where the latter stands relative to fully optimal SBPP solutions. If we take the totality of results into account, however, we have one case in 7n12s with a full demand matrix, where FIPP is within 3% of SBPP in the initial designs and matches SBPP when four working routes are adjusted. And in the larger test cases, the SBPP solution costs are all within the range of the gaps on the best feasible FIPP *p*-cycle designs. To this we should add the considerations from first principles that the most closely related scheme of all those considered to FIPP, is SBPP. These are the only two path-oriented, failure independent schemes above, i.e., path-type schemes that do not benefit from stub-release as does PR. All-told, therefore, it seems reasonable to expect that the FIPP architecture may be reasonably close in intrinsic efficiency to SBPP, just that we are not presently skilled enough at solving the FIPP *p*-cycle network design problems. In fact there is also one observation about FIPP and SBPP architectural properties that suggests that in at least one aspect FIPP could possibly be even more efficient than SBPP in some circumstances. Consider the following:

From one point of view it seems reasonable to surmise that a FIPP *p*-cycle could always be viewed as being formed from a specific choice of two backup routes, so that SBPP would have to serve as a lower bound for the spare capacity results of FIPP *p*-cycle designs. However, there is one important respect in which it can be demonstrated that the FIPP solution space is not simply a subset of the SBPP solution space. To explain this we need only look back at the case of the partially straddling path example in Figure 8-2(d). SBPP has a fundamental requirement that every working path is *fully* disjoint from its *own* backup route (except at its end-nodes), as well as fully disjoint from other working paths that share any spare capacity in their backup paths. There is no exception possible to this restriction under SBPP. But look again at Figure 8-2 (d). Here we see that under FIPP, a working path can in general have path segments in common with its own protection structure. The switching behaviour to allow this was explained when Figure 8-2 was introduced. What is seen in Figure 8-2 (d) is equivalent to a limited type of stub-release. Stub release is only otherwise found in true path restoration. In path restoration, restoration paths are allowed (when advantageous to the design) to re-use part of their own working paths within the protection path. In FIPP as we can see in Figure 8-2 (d), we can allow the backup path to be co-routed with the working path, thus effectively re-using the surviving component of the working route. This possibility separates the FIPP *p*-cycles architecture from SBPP and leaves it open that FIPP *p*-cycles might in some cases outperform SBPP in capacity efficiency, given suitable solutions to the optimization problem because the behavior in Figure 8-2(d) allows the optimizing design solver to occasionally save more capacity than would be possible under SBPP. The reader might also note that in the example where FIPP *p*-cycles wind up re-using their "stub" path segments, it is easy to construct an actual trap situation for SBPP. In SBPP if no backup route exists that is disjoint from the chosen working route then an infeasibility "trap" situation arises. In sparse networks this may mean that working routes tend to have to be significantly longer for SBPP feasibility. In contrast (as the example shows), FIPP does not fall into such traps and allows working paths to take shortest routes.

8.6.3. Inspection of FIPP Design Characteristics

Both as a form of validation of the functional correctness (mainly the property of 100% restorability) and to further portray and understand the FIPP architecture, we picked the 19 demand COST239 test case to draw out all the FIPP *p*-cycles that are in the solution. The completely restorable design was based on only five FIPP *p*-cycles shown in Figure 8-6 through Figure 8-10 below. Each figure shows one of the actual FIPP *p*-cycles chosen in the best-feasible design and the working demands associated with the cycle (shown using thin arrow-headed lines). We also show a simplified logical abstraction of each *p*-cycle and the end-to-end-node demand pairs that it protects.

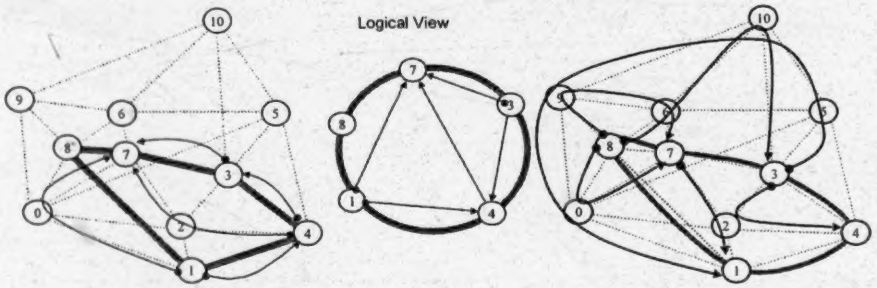


Figure 8-6. (a) FIPP *p*-Cycle A in Solution for Cost 239 network with 19 Demands (SCP), (b) logical view, (c) example of potential additional loading capability. (Adapted from [KoGrJLT])

Cycle *A* in Figure 8-6 protects up to *2* units of demand for demand relations *9,5,4,12,10*. The working routing is shown using thin arrow-headed lines. Notably if *p*-cycle *A* were to be used as a span protecting *p*-cycle its ratio of protected capacity to its own consumed capacity is only 1 because it has no straddling spans. However, on using the same cycle as a FIPP *p*-cycle, this ratio increases to 2.6 (end to end demands protected per channel-hop of the *p*-cycle) for the combination of demands shown in Figure 8-6(c), a much higher efficiency. The chances of cycle *A* actually being selected as a span *p*-cycle would therefore have been low, but this same cycle becomes of much higher merit as a FIPP *p*-cycle because it does have good straddling relationships when allowed to operate to protect paths solely on an end-to-end path basis. We note, therefore, that if we re-use the standard Apriori Efficiency [GrDoLEOS02] measure for span *p*-cycle selection to cut down the ILP problem size, we would tend not to include cycles such as this in the set of eligible cycles. This motivates further work on developing new heuristics and metrics for a priori cycle selection in FIPP. Another observation is that the number of demands that the FIPP *p*-cycle can protect can be increased, if the working routes are deviated from shortest path. This suggests not surprisingly that a joint model that simultaneously optimizes the working capacity and *p*-cycle placements may yield considerably more efficient designs by having even greater latitude over the group of compatible routes that are chosen to associate with each candidate *p*-cycle. The same FIPP *p*-cycle *A* can "soak up" more demands without any increase in spare capacity. This also motivates further work in developing a working routing strategy that maximizes FIPP efficiency.

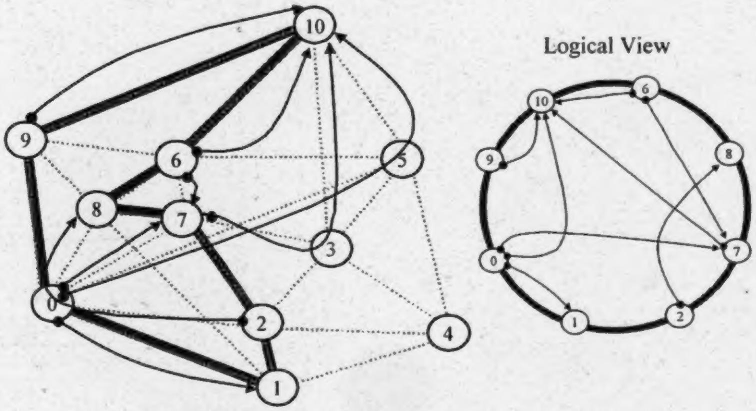


Figure 8-7. (a) *p*-Cycle (B) FIPP *p*-Cycle Solution for Cost 239 network with 19 Demands (SCP) and (b) logical view. (Adapted from [KoGrJLT])

Similarly, *p*-cycle *B* protects up to *2* units of demand for node pairs *8,2,3,18,17,16,15,0* in Table 8. FIPP *p*-cycle C protects demands on node pairs 9,2,18,16,15,12,11,10,1. FIPP *p-*Cycle *D* protects demands on node pairs *6,4,16,14* and FIPP *p*-cycle *E* protects demands on node pairs *7,4,18,13.*

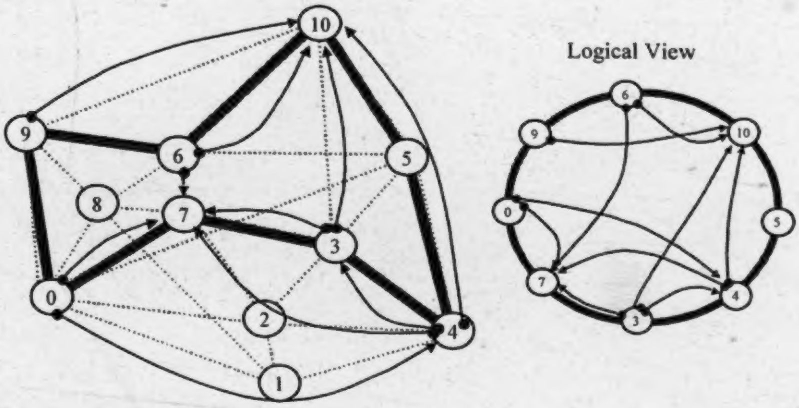

10
9
6
5
8
7
3
0
2
4
1
Logical View
6
9
10
0
5
7
3
4

Figure 8-8. (a) FIPP *p*-Cycle C and (b) logical view. (Adapted from [KoGrJLT])

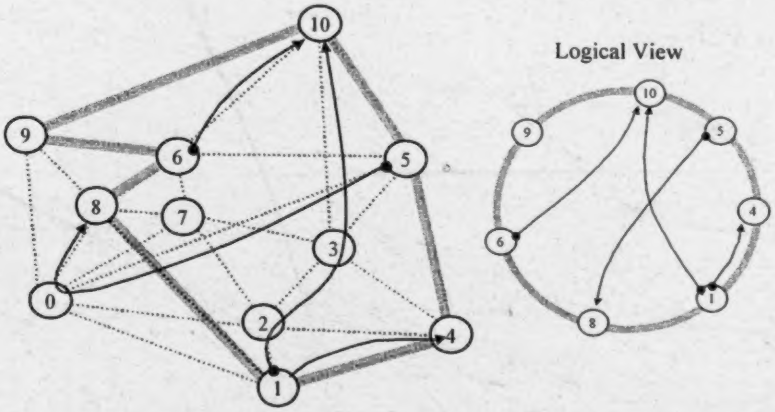

10
9
6
5
8
7
3
0
2
4
1
Logical View
10
9
5
4
6
1
8

Figure 8-9. (a) FIPP *p*-Cycle (D) and (b) logical view. (Adapted from [KoGrJLT])

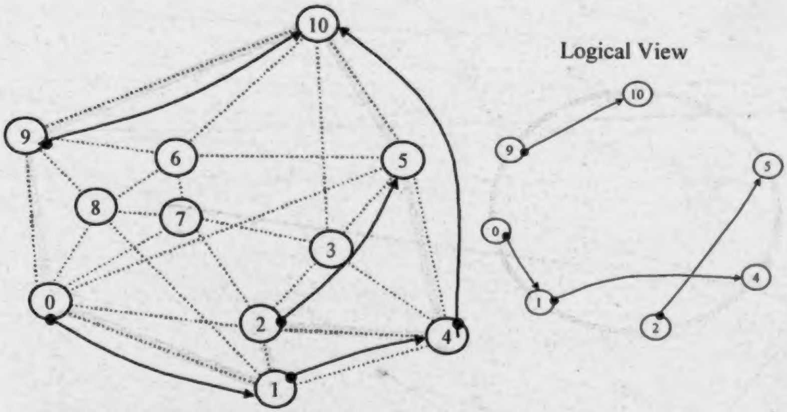

10
9
6
5
8
7
3
0
2
4
1
Logical View
10
9
5
0
4
1
2

Figure 8-10. (a) FIPP *p*-Cycle (E) and (b) logical view. (Adapted from [KoGrJLT])

On inspection of Figure 8-9 and Figure 8-10 we can see that the *p*-cycles do not give the impression of being very heavily loaded with protection relationships. In the present case, despite this seeming waste, the overall solution still requires less absolute spare capacity than the span *p*-cycle design. For interest and comparative purposes the optimal span protecting *p*-cycle solution for the same networks and set of working demands and routes is shown in Figure 8-11.

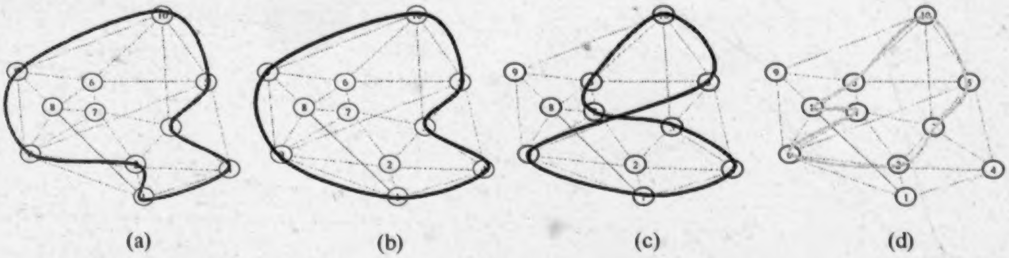

(a) (b) (c) (d)

Figure 8-11. Conventional Span *p*-cycle solution for COST239 with 19 demands and 1000 eligible cycles. (Adapted from [KoGrJLT])

By simple inspection it is not clear if there are any strategies that could be used to convert span *p*-cycles to FIPP *p*-cycles. We ran a quick test case where we limited the FIPP-SCP solver to choosing from the 4 span *p*-cycles from Figure 8-11. It did result in an optimal FIPP solution quickly but did not improve on the overall capacity efficiency relative to the span *p*-cycles design. There is no guarantee that a span *p*-cycle set can produce a feasible FIPP solution. This suggests, however, a possibly powerful strategy for assisting the ILP solution of the FIPP *p*-cycle network design problem: first solve the generally much easier regular *p*-cycle design problem and the corresponding PR problem and use its number of *p*-cycles and objective function values as guiding additional upper and lower bounds respectively for the FIPP *p*-cycle ILP solution.

8.7. Results for the FIPP-DRS-SCP Model

Experimental results show that using the FIPP-DRS ILP model herein, with the heuristic above to populate its candidate DRS set, we are able to produce FIPP *p*-cycle network designs that are more capacity-efficient than even strictly optimal conventional *p*-cycle network designs by as much as 6%, as shown in Figure 8-12. Each data point in Figure 8-12 corresponds to the optimally designed solution of the test case network with the indicated average nodal degree using the indicated design method. That is to say that the CPLEX terminations were optimal solutions for the problem models and data given to them. This is separate from absolute sub-optimality that may still be imputed from the restricted datasets given the corresponding problems. Although the FIPP *p*-cycle designs are the most sub-optimal by nature of the DRS-limiting heuristic used, they come within 10-18% of the capacity of SBPP network designs which have essentially no significant limitation on their global optimality (ILP termination and dataset provided). However, they provide the important property of offering a fully-pre-connected protection alternative. It should be noted that although the FIPP designs in 8.6, much more closely approached the capacity requirements of SBPP designs, the FIPP-DRS-SCP designs were solved in seconds or minutes, while the prior FIPP designs required up to 48 hours to solve.

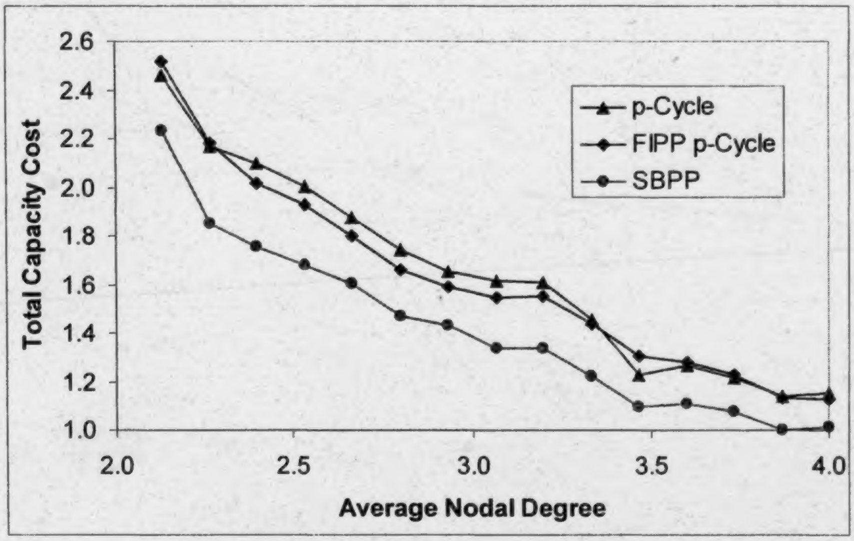


Figure 8-12 – Total capacity costs of FIPP *p*-cycle network designs using the FIPP-CDS ILP model. (Adapted from [KoGrDRNC05])

8.7.1. Improvements

Although the ILP-based heuristic above produced FIPP *p*-cycle designs that were highly efficient (i.e., more efficient than strictly optimal conventional *p*-cycle designs), the algorithmic enumeration of DRSs is not the best achievable. Consider that the number of DRSs enumerated in each test network above was 2205 (21 for each of the 105 demands), whereas the same test networks have possibly 6.6×10^{17} distinct DRSs in total. Obviously there is therefore scope for a DRS enumeration algorithm that will generate candidate DRSs in a more guided manner that should allow the FIPP-DRS model to produce designs closer in capacity to SBPP without increasing runtime.

One alternative DRS enumeration technique we tried was to limit the number of working routes per DRS. The rationale is that from an availability perspective, reducing the number of working routes protected by any one FIPP *p*-cycle will reduce the exposure to dual-failure combinations affecting two working routes within the same *p*-cycle and is a measure that would probably be insisted upon in later practice anyway. This is motivated by the work in [DoClONM03] on SBPP capacity-availability tradeoffs that showed that limiting the number of sharing relationships on any individual spare capacity channel will not significantly affect the network efficiency, but will considerably improve its availability. The corresponding limitation for FIPP *p*-cycles is to restrict the number of working routes sharing any FIPP *p*-cycle, or in other words, limit the maximum possible size of each DRS. A simple modification of the ***`GenerateDRSs`*** algorithm was made by adding another "if" statement stating that "`DRS c is complete`" if it already includes a specified number of working routes. Then using one of the test case networks (the 22-span member of the test network family), we ran the FIPP-DRS problem with numerous different restrictions on DRS size ranging from 2 through 10.

Like the previous results discussed, the FIPP *p*-cycle designs had efficiencies approximately 6% below an optimally designed *p*-cycle network and 10% above an optimally designed SBPP network. Each data point in Figure 8-13 corresponds to the normalized capacity cost of the FIPP-DRS designed test network with the indicated maximum number of working routes per DRS. The solid (green) horizontal line intersecting the y-axis at approximately 1.16 corresponds to *p*-cycle capacity cost for that same network. We can see that so long as we allow 5 or more working routes per DRS, the exact number will have little impact on capacity efficiency. Similar effects were seen in [DoClONM03] with regards to SBPP sharing limits. Since the set of eligible DRSs provided to the ILP model are enumerated randomly, we also investigated the sensitivity of the resulting network designs to the specific set of DRSs provided. To do so, we ran 10 instances of the previous test data (i.e., 10 instances of the 22-span test case network with a maximum of two disjoint routes per DRS, another ten with three disjoint routes allowed per DRS, etc.), with the average of all ten instance for each scenario shown in Figure 8-13. When the worst-case and best-case results are extracted for each scenario (i.e., for each specified maximum number of working routes per DRS), we actually find that the solution quality is quite robust with regard to specific set of DRSs provided: solutions ranged little more than ±0.5% over all test cases. In fact, for most cases, the error bars were lost in the data point markers when plotted on top of the curve in Figure 8-13. While this suggests that any particular solution obtained with the heuristic is quite repeatable (in the sense of equivalently good designs being reached, but which may differ in detail). But it also implies that finding a DRS enumeration method that produces any significantly better results may be difficult.

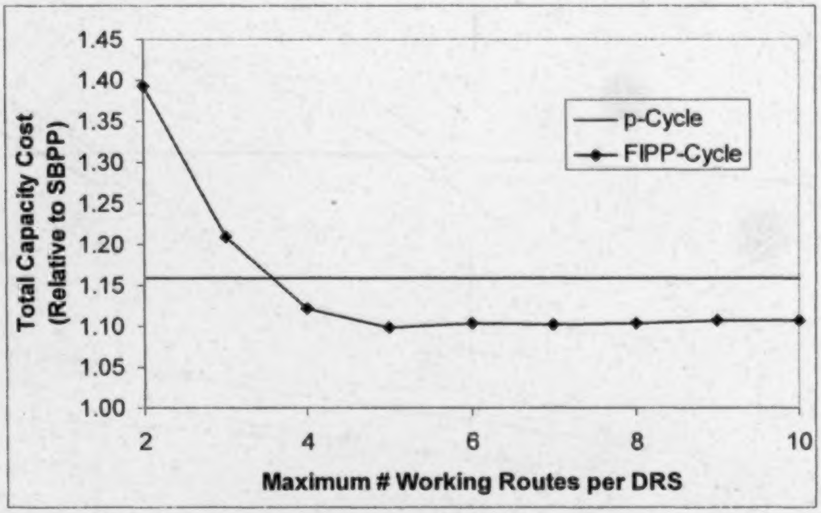


Figure 8-13 – Total capacity costs of FIPP *p*-cycle network designs with limited numbers of demand relations per CDS. (Adapted from [KoGrDRCN05])

8.8. Results for the FIPP-JCP-DRS Model

8.8.1. Varying Cycles Per DRS

In this experiment, we varied the number of eligible cycles provided as input to the solver. All other parameters are the same as the standard parameters in 8.5.3.2. Figure

8-14 shows the corresponding design cost profile. In both cases the *X* axis shows the number of eligible cycles per DRS.

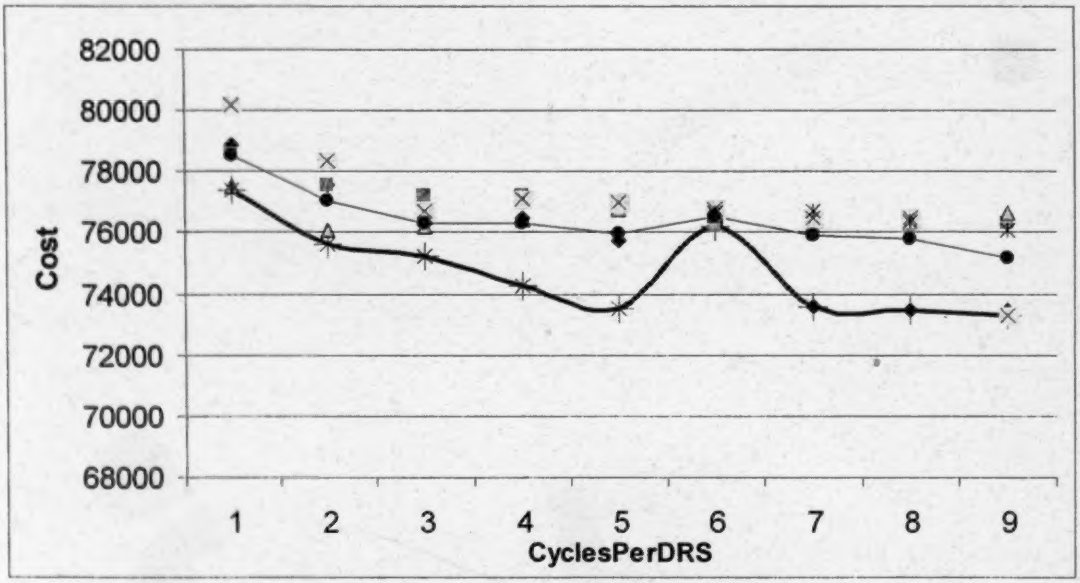


Figure 8-14. Cost Versus Cycles per DRS. (Adapted from [KoBaPNET06])

Standard redundancy for this design varies between 55 and 75%. Standard redundancy does not take into consideration distance effects, as it is purely a capacity calculation. In a joint design, especially, spare to working channel redundancy is actually not an effective metric because the working routing may deviate from the shortest path and thus may provide an artificial lower redundancy by increasing the network working capacity. The total network cost however shows a more accurate profile. What we see in the lower thicker line in Figure 8-14 is that the costs show a saturation trend as the number of eligible cycles per DRS are increased. Beyond five cycles per DRS cost improvements are only marginal. This shows the existence of a threshold effect on the number of eligible cycles that need to be provided to the solver. As long as five eligible shortest cycles are provided for each DRS, we can be sure that the no further significant cost reductions will result from additional cycles. This results in a reduction in problem size as there are only 5*(Number of DRSs) eligible cycles provided to the solver.

8.8.2. Varying Maximum Number of Routes In Each DRS

In this experiment, we varied the maximum size of the DRS from 2 routes to 16 routes per DRS, while keeping all the other parameters as given in section 8.5.3.2. From the resulting cost profile in Figure 8-15 we can see that the average cost saturates at around 8 to 10 routes per DRS. The least expensive design is when a maximum of 10 routes are allowed per DRS. In practice a DRS with 10 routes will already be large enough to reach limiting efficiency. As an effective analogy, this is similar to the threshold number of sharing relationships a single spare channel needs to support in SBPP to achieve limiting efficiency as discussed in [DoClONM03].

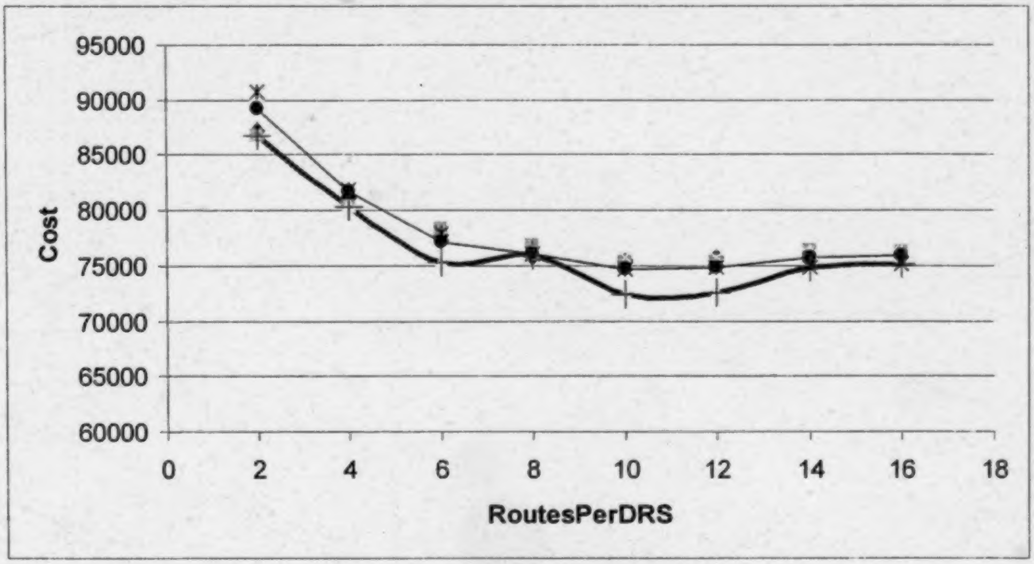


Figure 8-15. Cost versus Routes Per DRS. (Adapted from [KoBaPNET06])

8.8.3. Varying The Number of Eligible Routes Per Demand

In this experiment the number of eligible routes per demand is varied between 1, the single shortest route, and 8 shortest routes. In Figure 8-16 we can see that as the number of eligible routes per demand is increased from one to two, there is a decrease in the total network cost. However as the number of eligible routes increases beyond two, there is a sharp increase in the total cost of the network as seen by the thin ascending line. This seems counter intuitive as increasing the number of eligible working routes should not, in any case, worsen the solution. In other words, a case with 2 eligible working routes per demand is only a subset of 3 eligible working routes per demand case. Indeed the best solution obtained with the 2 eligible routes can be used as the best solution over all cases, as indicated by the thick rigid line in Figure 8-16, if increasing the number of eligible routes worsens the solution.

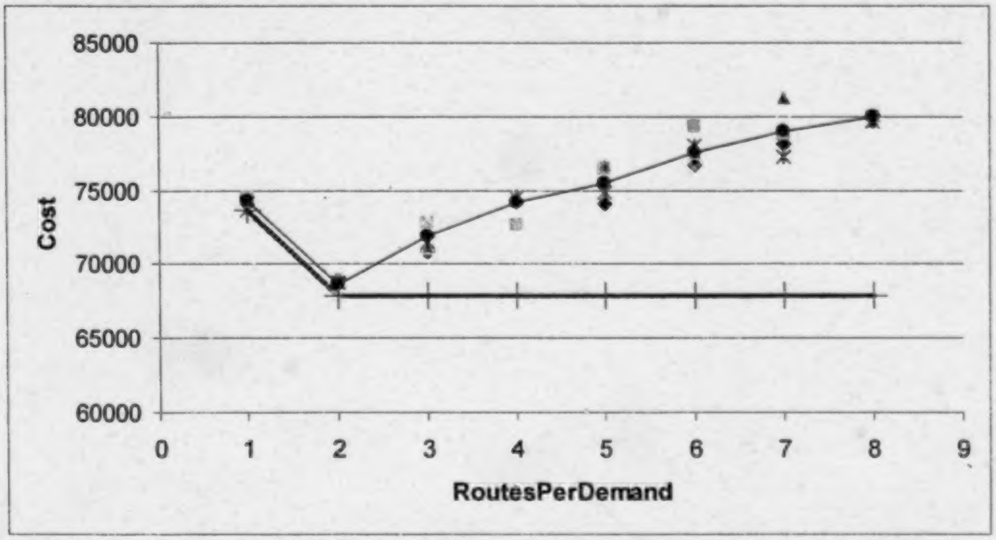


Figure 8-16. Cost versus number of eligible routes per demand. (Adapted from [KoBaPNET06])

To understand the reason for this discrepancy, we examined the eligible route set generated in the COST239 test network. In Figure 8-17 we see a 95th percentile average length profile for the set of routes included in the Nth shortest route set.

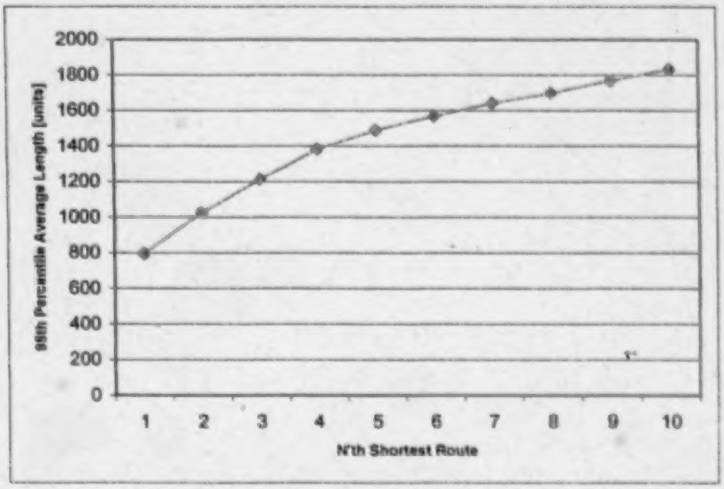


Figure 8-17. Average Length versus Nth shortest route. (Adapted from [KoBaPNET06])

To generate this graph we enumerated the *n* shortest routes for each possible O-D in the COST239 network graph by length. We then separated the routes into groups corresponding to the rank of their relative length with respect to the shortest route from the same OD pair. The top 5% of the routes (by length) were discarded and an average length was calculated based on the remaining routes. What we can see in Figure 8-17 is that as the number of eligible routes per O-D pair is increased, the average length of the set of working routes also goes up significantly. As an example, the average route in the set of 3rd shortest working routes is 50% longer than the average route among the shortest working routes. The GenerateDRS algorithm is essentially a random algorithm that selects routes from the master set to be part of the DRS. When this master set is populated with an extremely large number of long routes (as compared to the number of short routes), it becomes more likely that the limited sample set of eligible DRSs generated, contain, on average, longer routes. Since there is no restriction in GenerateDRS that all enumerated routes be part of at least one eligible DRS, the routes that are in the solution from the "2 eligible routes per demand" trial have a much poorer chance of being efficiently placed into DRSs generated from the "3 eligible routes per demand" trial. The problem worsens as the size of the master eligible route set increases. On average, as the number of eligible routes per demand is increased, the average length of the working routes protected by the generated DRSs will increase as well. This leads to degradation in design efficiency and an increase in network costs as the working capacity is routed over longer paths and the FIPP *p*-cycles also, on average, have to be larger.

To verify these results we re-ran the experiment with a slight modification to include a DRS set history. In other words in the test case with the 3 shortest eligible routes, we also included in the input all the DRSs generated as part of the 2 shortest route test case and all the DRSs generated as part of the single shortest route test case. In general, the test case where n-shortest routes were provided contained all the set of DRSs generated when (n-1), (n-2) ... 1 eligible route(s) per demand are provided. This guarantees that, in the worst case, the solver in the n-eligible route case can always choose any of the previous solutions as optimal or in other words, the design cannot worsen as the number of eligible working routes per demand are increased. Results for this trial are in Figure 8-18. Figure 8-18 shows the total network cost, cost of working and cost of spare as the number of eligible working routes per demand is increased. In Figure 8-18 we see that the joint design is 13% more cost efficient than the non-joint design (i.e. the difference between the first and second data point). The standard cost redundancy improves to 53% for the joint case. It can be seen that by choosing some alternate working paths, as seen by the increase of the working cost, the solver is able to greatly reduce the spare capacity cost. What we also see in Figure 8-18 is that once all routes up to and including the second shortest route per demand are enumerated, no more gains are to be had in efficiency by enumerating more routes. In other words, at the point where all routes up to the second shortest route are enumerated, limiting efficiency is reached. The existence of this threshold validates our hypothesis from section 8.4.2.1 where we speculated on the existence of a threshold number of eligible routes. But one question still remains. Why does the threshold exist at all and why does it exist at the specific point “2” for this network? To understand and answer this question we go back to Figure 8-17 and see that the average route length of the set of second shortest routes was 26% longer than the average shortest route. The length difference between the average lengths of the shortest and 3rd shortest routes is 50%. Recalling our discussion in section 8.4.2.1 earlier: To offset a 50% increase in the cost of a single working route, in an approximately 50% redundant survivable network, the corresponding spare capacity cost reduction must be, relatively, at least a 100%. In other words, with our current experimental assumptions of geographic distance being approximated as cost –if a working route that is 10% longer than the shortest route is chosen, the corresponding spare capacity savings must actually be at least 20%. This level of improvement is extremely unlikely to happen. The reason the solver sees a benefit from going from one eligible route to two per demand must be that it is now able to spread out the working capacity among essentially equal shortest routes. To verify these, we drew out the solutions for the first two data points in Figure 8-18. The results are in Figure 8-19 and Figure 8-20. In Figure 8-19 (and Figure 8-20), the FIPP *p*-cycle is shown using the thick line and the working routes are shown using the thin arrow tipped lines. We can see that when only the shortest routes are allowed as input, the complete solution requires building 7 FIPP *p*-cycles (6 unique cycles). When the second shortest route is introduced, the number of FIPP *p*-cycles is reduced from 7 to 4, and 6 working routes are chosen from the set of second shortest working routes (shown by the dotted arrow tipped line.) What is important to note however is that the newly included routes are no more than a few percent longer than the previous solution. From Figure 8-18 we know that including the third shortest route is of little benefit because, on average, the third shortest working route is much longer. The joint design of FIPP *p*-cycles works because the solver can now cut down the number of FIPP *p*-cycles built by routing the working along multiple equally short working routes which is a kind of working capacity leveling over multiple routes. It is not because the solver is allowed to choose very long routes. Based on this insight, as long as all routes within say 20% of the shortest route length are included in the input eligible working route set, maximal efficiency is reached.

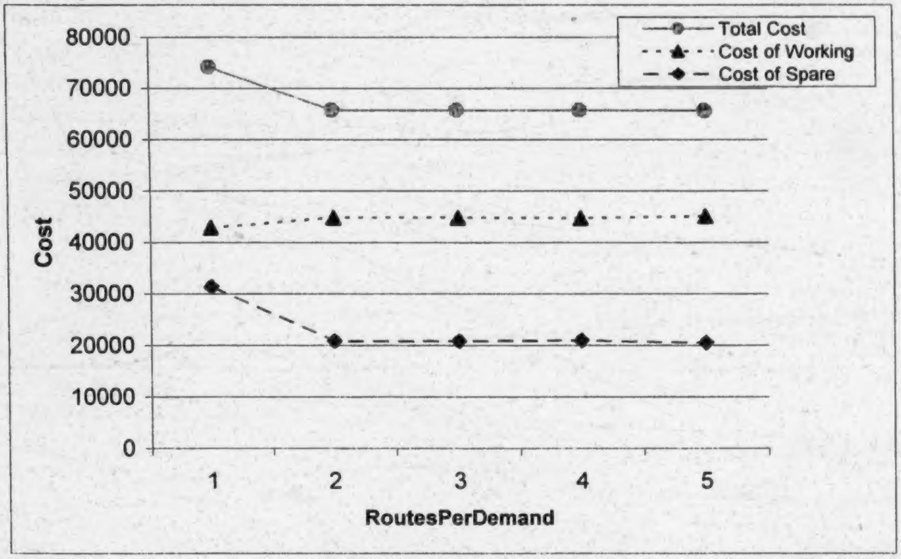


Figure 8-18. Number of Eligible Routes per Demand vs. Cost, with history propagation. (Adapted from [KoBaPNET06])

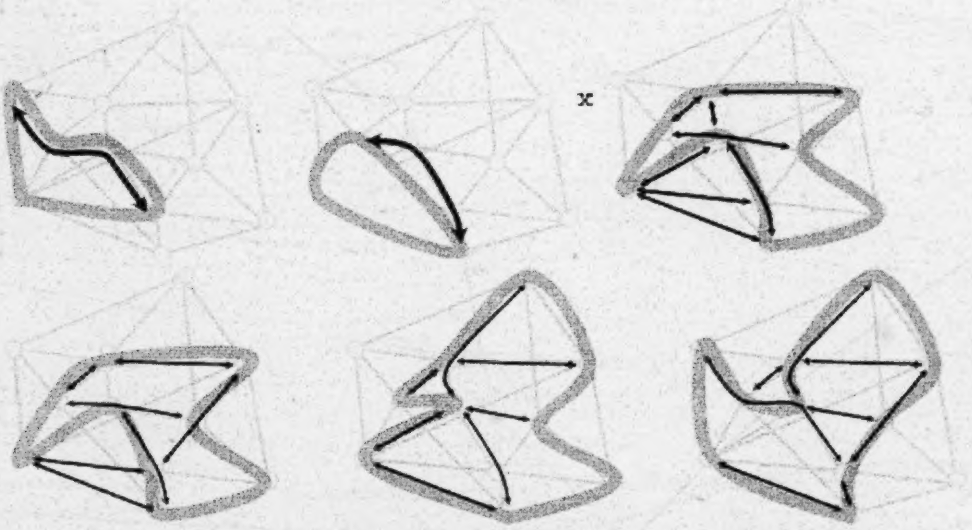


Figure 8-19. Complete FIPP DRS Solution with Shortest Routes Only. (Adapted from [KoBaPNET06])

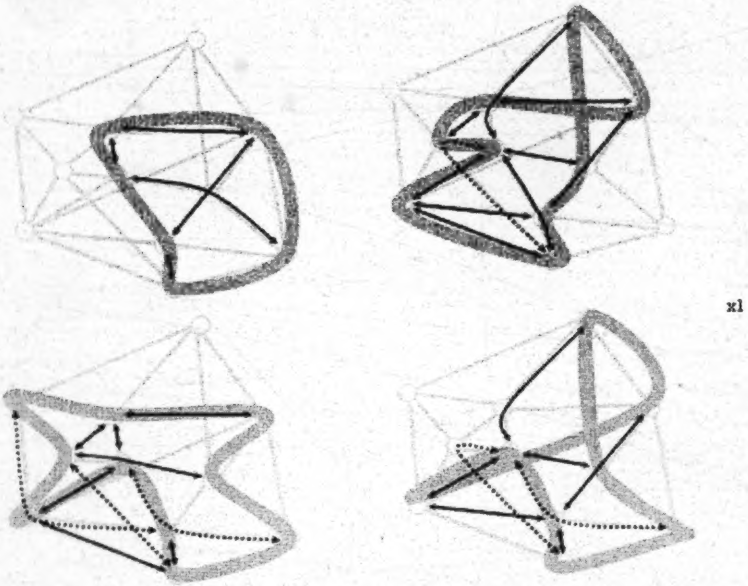


Figure 8-20. Complete FIPP solution when up to two shortest routes are allowed. (Adapted from [KoBaPNET06])

8.9. Summary

In this Chapter we have proposed many design approaches for FIPP *p*-cycle design and presented results. What is significant about FIPP *p*-cycles is that they offer all the desirable features of SBPP, namely failure independent end-node fault detection and activation of protection paths, built-in protection against either node or span failures and dynamic path-restoration like capacity efficiency, but without requiring any real-time cross-connection between optical channels to form protection paths in real time. In addition, they closely match SBPP in capacity efficiency. The protection structures from which protection paths are claimed upon failure are fully pre-connected structures which mean they can be in completely tested and monitored "known-good" states prior to their emergency use upon some network failure. In all other schemes of this characteristic range of spare capacity efficiency, the efficiency is ultimately achieved by resorting to on-demand switched assembly of spare channels into backup wavelength paths. Only after formation following the start of the emergency will it be found out if these paths work or not.

Further work on FIPP *p*-cycles is currently in progress at the Network Systems Group at TRLabs and is summarized briefly in the following Chapter.

Chapter 9. CLOSING DISCUSSION

9.1. Summary of Thesis

In this thesis our main intent is to provide network planners and operators with better tools, knowledge and strategies to design *p*-cycle-based optical networks. Chapter 1 introduced the topic area. In Chapter 2 we presented an overview of Transport Networking Fundamentals. We looked at nodal equipment architecture, transmission systems and technologies like SONET, CWDM, DWDM, etc. We also looked at various physical and logical layer techniques for network restoration such as Rings, APS and *p*-cycles, with pointers to extensive reference material for further information. We also reviewed some topics from the recent literature that deals with pre-connection and optical layer protection.

Sets, graphs and optimization concepts were reviewed in Chapter 3. We also reviewed routing algorithms for route and cycle finding. Advanced topics like modularity and economy of scale were briefly discussed. Chapter 4 contained a network planner's appreciation of some issues in the physical layer (optical link design), where we presented a detailed discussion about the steps involved in activating a single point-to-point optical link. We also discussed pre-connection and failure independence and explained their importance in optical networking. Our aim was to help the designer and the general research community understand why the assumption that on-the-fly concatenation of light-path channels to form backup paths after the failure, in real-time, is invalid, given the current generation of optical networking technologies. Power levels, error rates, amplifier gains, etc. must all be tuned just right, and often manually, end-to-end, for the point-to-point link to work.

Chapter 5 detailed our first study topic, which was on mining rings to *p*-cycles. Two specific design models were presented, one where the problem was to maximally serve demands using existing resources mined to *p*-cycles whereas the other was to selectively and strategically add the minimum capacity required to serve a specified uniform demand growth multiplier. Tests on 17 pseudo-random ring networks showed significant potential for migration. Subsequent tests on real network data from TELUS, for their Calgary metro network, showed that we could support a uniform demand multiplier of 1.5 without any capacity addition. To literally double the carrying capacity of the TELUS network, only a few additional modules of capacity need to be added. We also found that restricting the re-routing of existing working demand reduced the growth potential slightly, but could be considered a valid trade-off as the operator can migrate his network without affecting his customers. Overall we established that *p*-cycles are almost as capacity efficient as mesh networks for ring-mining purposes. The nodal compatibility of *p*-cycles and ring networks may eventually make *p*-cycles a better choice for a migration target.

In Chapter 6 we presented our study on *p*-cycle path lengths. We found that a threshold hop limit effect does exist. In our test networks, we found that the hop limit was at around 6 to 9 hops for *p*-cycles. This is similar to what mesh networks exhibit, except that the hop limit thresholds were about 3 hops lower than the corresponding path length limited *p*-cycle design. We also proved, for our test networks, that the extra ability to precisely control path lengths does not actually produce any better results than the standard *p*-cycle design model. The more complex path length limiting model may actually be replaced with a simple circumference limited model without any penalty.

In Chapter 7 we presented our work on supporting multiple Quality of Protection classes using the same set of resources in a *p*-cycle network. We showed, using examples, how a multi-QoP framework can be efficiently implemented in a static *p*-cycle network. We also discussed how provisioning of dual failure survivable platinum services can be integrated with the other survivability classes. The results showed that with the straddling routing principle for platinum service channels, a significant level of dual failure protection may be obtained for free, with just the resources required for a single failure protected network. With the introduction of economy channels, we showed how a network may be operated with almost no spare capacity. This enabled the operator to run a network where almost every single in-use channel generated some revenue. We also confirmed other results in the literature that have shown that a fully dual failure survivable network is around 300% redundant.

In Chapter 8 we presented a significant extension to the existing body of knowledge and techniques for *p*-cycles: FIPP *p*-cycles. FIPP *p*-cycles are almost as capacity-efficient as SBPP and PR, while at the same time providing failure independent and pre-connected optical protection paths. FIPP *p*-cycles are therefore the first efficient, failure independent, pre-connected, path protecting mechanism in the literature. We discussed both joint and non-joint designs and also presented a heuristic approximation for the DRS-based FIPP *p*-cycle design model.

To summarize, the main contributions from this thesis are listed in the following section.

9.2. Main Contributions

There are five main areas of contributions of this thesis:

1. Conception and development of FIPP *p*-cycles.
    - Design of FIPP *p*-cycles using the cycle – route set selection principle.
    - DRS-based FIPP *p*-cycle design.
    - Joint design of FIPP *p*-cycles.
    - FIPP *p*-cycle network operation techniques.
    - DRS-based heuristics for FIPP *p*-cycle design.
    - Detailed review and comparison of FIPP *p*-cycles to SBPP.
2. Ring-mining to *p*-cycles.
    - Migrating to *p*-cycles using existing ring nodal equipment.
    - Ring-mining where existing working routes can be re-routed.
    - Migrating a network without affecting current working traffic.
    - Selective minimum capacity addition to support specified demand growth multiplier.
3. Design of *p*-cycles with multiple qualities of protection classes.

    - Operational paradigm for supporting multi-QoP using the same set of resources.
    - Dual failure protection strategies using static *p*-cycles: straddler routing and dispersal principles.
    - Operating networks without any redundant capacity by selective placement of pre-emptible economy channels.
    - Effect of traffic growth on network efficiency in a multi-QoP network.
    - Strategies to maximize revenue from a network by correctly distributing a multi-QoP mix.
4. Dual failure survivability for selected services using static *p*-cycles.
    - Static routing techniques to ensure dual failure protection.
    - Operational paradigm to support dual failure survivability by preemption of gold protection by failed platinum channels.
5. Designing *p*-cycles with explicit path length restrictions.
    - Explicit path length limitations instead of circumference limits.
    - Demonstration of the existence of a threshold hop limit effect.
    - Establishing the trade-off in terms of higher thresholds for pre-connected-ness.
    - Bi-criteria techniques to obtain shorter protection paths.

9.3. Publications and Presentations

As discussed at the beginning of Chapters 5 to 8, the work presented in this thesis resulted directly, or in part, in the production of two published journal articles, one book Chapter, five articles published in various IEEE conference proceedings and one patent filing on FIPP *p*-cycles. Numerous related presentations were also made locally and internationally. We have also recently submitted two more journal submissions based on the work in this thesis. These publications, listed in each Chapter, are listed below as a summary:

1. W.D.Grover, **A. Kodian**, "Failure Independent Path Protecting *p*-Cycles," provisional US Patent filed Nov. 29, 2004.

2. **A. Kodian**, W. D. Grover, J. Slevinsky*, D. Moore* (*TELUS Communications), "Ring-Mining to *p*-Cycles as a Target Architecture: Riding Demand Growth into Network Efficiency," *Proceedings of the 19th Annual National Fiber Optics Engineers Conference (NFOEC 2003)*, Orlando, September 2003, pp.1543-1552.

3. **A. Kodian**, A. Sack, W. D. Grover, " *p*-Cycle design with hop limits and circumference limits," *Proceedings of IEEE BROADNETS 2004*, San Jose, October 2004. **Winner of IEEE- Phillips best paper award**.

4. W.D. Grover, J.E. Doucette, **A. Kodian**, D. Leung, A. Sack, M. Clouqueur, G. Shen, "Design of Survivable Networks based on *p*-cycles," Handbook of Optimization in Telecommunications," Kluwer Academic Publishers, Editors: Panos M. Pardalos, Mauricio G. C. Resende.

5. **A. Kodian**, W.D. Grover, "Failure independent path-protecting *p*-cycles: efficient and simple fully pre-connected optical-path protection," *IEEE Journal of Lightwave Technology*, vol. 23, no.10, October 2005

6. **A. Kodian**, A. Sack, Wayne D. Grover, "The threshold hop-limit effect in *p*-cycles: Comparing hop- and circumference-limited design," invited publication in *Journal of Optical Switching and Networking*, Elsevier, July 2005, pp. 72-85.

7. **A. Kodian**, W.D. Grover, "Multiple-Quality of Protection Classes Including Dual-Failure Survivable Services in *p*-Cycle Networks," in *Proceedings of IEEE BroadNets2005*, Boston, October 3-7, 2005.

8. **A. Kodian**, W. D. Grover, J. Doucette, "A Disjoint Route Sets Approach to Design of Failure-Independent Path-Protecting *p*-Cycle Networks," in the 5th International Workshop on *Design of Reliable Communication Networks (DRCN 2005)*, Ischia (Naples), Italy, 16-19 October 2005.

9. W.D. Grover, **A. Kodian**, "Failure-Independent Path Protection with *p*-Cycles: Efficient, Fast and Simple Protection for Transparent Optical Networks," invited paper in *Proc. IEEE International Conference on Transparent Optical Networks*, July 3-7, 2005, Barcelona, Spain.

9.3.1. Presentations/Reports/In Preparation

10. **A. Kodian**, W.D. Grover, "How do Survivable Networks Work: How to make a profit and ensure network availability despite daily failures?" Campus Computing 2005, Edmonton, Alberta, June 21, 2005.

11. **A. Kodian**, D. Baloukov, W.D. Grover, "Joint optimization of working and spare capacity in FIPP *p*-cycle network design," to be submitted to Photonic Network Communications.

12. **A. Kodian**, W.D. Grover, "*p*-Cycle Network design with multiple Quality of Protection and dual failure survivability," to be submitted to Journal of Optical Switching and Networking.

13. **A. Kodian**, W. D. Grover, "Ring-Mining to *p*-Cycles as a Target Architecture," TRLabs Tech Forum 2003, Calgary, Alberta, October 2003. (Presentation)

14. **A. Kodian**, W. D. Grover, "Operating a completely static *p*-cycle network with Multiple Quality of Protection Service Classes, including dual failure protection of selected services," TRLabs Tech Forum 2004, Saskatoon, Saskatchewan, October 2004. (Presentation)

9.4. Topics for Further Research

This section outlines some research ideas generated during the studies in Chapters 5 to 8, but never actually implemented. Some of these ideas are currently being researched by newer members of our research group.

9.4.1. Envelope-Based Operation of FIPP *p*-Cycle Networks

In span restoration and *p*-cycles, it is the working capacity on a failed span that are protected and restored. In FIPP *p*-cycles it is the individual working wavelengths that is protected. A new operating paradigm, called the Protected Working Capacity Envelope (PWCE), was proposed by Grover in [Grov03] with its corresponding application to *p*-cycles in [ShGrKLUWER04]. In operating a network under the PWCE framework, as defined for span restoration, if the network is properly designed so that pre-determined amounts of working capacity are fully restorable over all single span failures, then any working lightpaths that are routed over these channels are also restorable by extension. The predetermined amounts of working capacity is considered to be an envelope of protected capacity within which lightpaths can be dynamically provisioned or taken down, without explicitly providing for their protection. The PWCE concept improves on SBPP. In SBPP, as we saw in Chapter 2, the backup route must be found at the time of provisioning the working path and that the backup path be cross connected in real-time after the failure. This requires a global database that maintains information about every connection that arrives or departs. In contrast in the PWCE concept, only the working path is cross connected in real time. For more information on the PWCE concept we refer the reader to [Grov03].

FIPP *p*-cycles lend themselves to PWCE-type operation in an even more attractive way because entire working paths between given O-D pairs will become pre-defined structurally protected entities. First, no explicit cross-connections or calculations need to be made for protection when a lightpath demand arrives, between a given O-D pair, as long as a FIPP *p*-cycle is available. Second, and unlike span protection based PWCE, no cross-connections are required even when working demand is to be provisioned. One can then imagine a network that is built essentially out of pre-made cross-connections where end-to-end demands may arrive and depart frenetically, but the network as a whole requires no cross-connections in the demand arrival time-scale. In a network that operates under this paradigm, there is no more 'routing' just switching at the end-nodes of the failed working lightpath. The only cross-connections are when the envelope is to be resized but this is an essentially off-line process.

9.4.2. Multi-QoP in FIPP *p*-Cycle Networks

This is a fairly simple extension of the multi-QoP framework proposed in Chapter 7 for FIPP *p*-cycle network operation. The key new piece will be the creation of strategies specific to FIPP *p*-cycles for dual failure protection of platinum working channels.

9.4.3. Dual Failure Analysis of FIPP *p*-Cycle Networks

The basic idea is to develop a design model that takes as input a R1=100% design and analyses it for maximal R2. In other words, the idea is to answer the question: "How many demands in the network are protected against dual span failures?"

9.4.4. Ring Mining to FIPP *p*-Cycles

Mining rings to FIPP *p*-cycles is a straightforward extension of the work in Chapter 5. We found that most ring networks exhibited high levels of balance and capture. In other words, most demands originated and terminated on the same ring. Converting rings to FIPP *p*-cycles is an even easier step than converting to span *p*-cycles, with the benefit that demands can now be protected end-to-end.

9.4.5. Simplifying FIPP ILP-based Design Using Column Generation

In the FIPP-DRS model in Chapter 8, we can see that the main complexity of the design model is in enumerating the set of disjoint routes. As the number of enumerated DRSs is increased, the run-time and complexity of the model increases. This is particularly problematic in joint design of FIPP *p*-cycles. Column generation is an optimization technique where variables are enumerated on demand and therefore the explicit enumeration of DRSs is not required. This approach promises to simplify optimal models published in Chapter 8. Attempting to speed up the FIPP design models is currently being explored as a collaborative opportunity between Dr. Bridgitte Jaumard of Concordia University and the Network Systems Group at TRLabs.

References

[1] [Grov03] W. D. Grover, Mesh-Based Survivable Networks. Upper Saddle River, New Jersey, USA: Prentice Hall PTR, 2003.

[2] [GrStICC98] W. D. Grover, D. Stamatelakis, "Cycle-Oriented Distributed Preconfiguration: Ring-like Speed with Mesh-like Capacity for Self-planning Network Restoration," Proceedings of IEEE International Conference on Communications (ICC 1998), Atlanta, GA, pp. 537-543, 7-11 June 1998.

[3] [GrStDRCN00] W. Grover, D Stamatelakis, "Bridging the ring-mesh dichotomy with *p*-cycles", Proc. IEEE / VDE DRCN 2000, Munich, Germany, April 2000, pp. 92-104.

[4] [DoGrNFOEC03] J. Doucette, W. D. Grover, "Node-Inclusive Span Survivability in an Optical Mesh Transport Network," Proceedings of the 19th Annual National Fiber Optics Engineers Conference (NFOEC 2003), Orlando, September 2003, pp. 634-643.

[5] [DoGrDRCN01] J. Doucette, W. D. Grover, "Comparison of Mesh Protection and Restoration Schemes and the Dependency on Graph Connectivity," Proceedings of the 3rd International Workshop on Design of Reliable Communication Networks (DRCN 2001), Budapest, Hungary, pp. 121-128, October 2001.

[6] [KoGrNFOEC03] A. Kodian, W. D. Grover, J. Slevinsky, D. Moore, "Ring-Mining to *p*-Cycles as a Target Architecture: Riding Demand Growth into Network Efficiency," Proceedings of the 19th Annual National Fiber Optics Engineers Conference (NFOEC 2003), Orlando, September 2003, pp.1543-1552.

[7] [ShGrICC03] G. Shen, W.D. Grover, "Extending the *p*-Cycle Concept to Path-Segment Protection," Proc. IEEE International Conference on Communications (ICC 2003), Anchorage, May 11-15, 2003, Session ON3, pp.1314-1319.

[8] [ShGrJSAC03] G. Shen, W. D. Grover, "Extending the *p*-cycle concept to path segment protection for span and node failure recovery," IEEE Journal on Selected Areas in Communications, vol. 21, no. 8, Oct. 2003, pp. 1306-1319.

[9] [ShGrKLUWER04] G. Shen, W. D. Grover, " Design of Protected Working Capacity Envelopes Based on *p*-Cycles: An Alternative Framework for Survivable Automated Lightpath Provisioning," in Performance Evaluation and Planning Methods for the Next Generation Internet, A. Girard B.Sanso, F.Vazquez-Abad (editors), GERAD 25th Anniversary Series, Springer, 2005, pp.63-97.

[10] [ShGrAPOC04] G. Shen, W. D. Grover, "Performance of protected working capacity envelopes based on *p*-cycles: Fast, simple, and scalable dynamic service provisioning of survivable services," Proc. Asia-Pacific Optical and Wireless Communications Conference (APOC) 2004, Beijing, China, 7-11 Nov. 2004, vol. 5626.

[11] [ChChTON04] T.Y.Chow, F.Chudak, A.M. Ffrench, "Fast Optical Layer Mesh Protection Using Pre-Cross-Connected Trails," IEEE/ACM Transactions on Networking, Vol 12, No. 3, June 2004, pp 539-547.

[12] [LiRuICCCN04] C. Liu, L. Ruan, "Finding good candidate cycles for efficient *p*-cycle network design," Proc. 13th International Conference on Computer Communications and Networks (ICCCN 2004), Chicago, Illinois, USA, 11-13 Oct. 2004, pp. 321-326.

[13] [ShMaISCC04]S. Shah-Heydari, W. Mardini, O. Yang, "Theoretical analysis of restorability of Hamiltonian protection cycles in random mesh networks," Proc. Ninth IEEE Symposium on Computers and Communication (ISCC 2004), Alexandria, Egypt, 28 Jun.-1 Jul. 2004, pp. 921-926.

[14] [SaGrNETWORK04]A. Sack, W. D. Grover, "Hamiltonian *p*-cycles for fiber-level protection in homogeneous and semi-homogeneous optical networks," IEEE Network, vol. 18, no. 2, Mar./Apr. 2004, pp. 49-56.

[15] [HuCoCOMMAG02] H. Huang, J. A. Copeland, "A series of Hamiltonian cycle-based solutions to provide simple and scalable mesh optical network resilience," IEEE Communications Magazine, vol. 40, no. 11, Nov. 2002, pp. 46-51.

[16] [ScONDM04] D. A. Schupke, "An ILP for optimal *p*-cycle selection without cycle enumeration," Proc. Eighth IFIP Working Conference on Optical Network Design and Modelling (ONDM 2004), Ghent, Belgium, 2-4 Feb. 2004.

[17] [DoHeDRCN03] J. Doucette, D. He, W. D. Grover, O. Yang, "Algorithmic approaches for efficient enumeration of candidate *p*-cycles and capacitated *p*-cycle network design," Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), Banff, Alberta, Canada, 19-22 Oct. 2003, pp. 212-220.

[18] [KaReDRCN03] J. Kang, M. J. Reed, "Bandwidth protection in MPLS networks using *p*-cycle structure," Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), Banff, Alberta, Canada, 19-22 Oct. 2003, pp. 356-362.

[19] [GrICT03] C. G. Gruber, "Resilient networks with non-simple *p*-cycles," Proc. Tenth International Conference on Telecommunications (ICT 2003), Papeete, Tahiti, French Polynesia, 23-28 Feb. 2003, pp. 1027-1032.

[20] [GrDoLEOS02] W. D. Grover, J. Doucette, "Advances in optical network design with *p*-cycles: Joint optimization and pre-selection of candidate *p*-cycles," Proc. IEEE LEOS Summer Topicals 2002, Mont Tremblant, Québec, Canada, 15-17 Jul. 2002, pp. 49-50.

[21] [StGrTON00] D. Stamatelakis, W. D. Grover, "Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles ('*p*-cycles')," IEEE Transactions on Communications, vol. 48, no. 8, Aug. 2000, pp. 1262-1265.

[22] [StGrJSAC00] D. Stamatelakis, W. D. Grover, "IP layer restoration and network planning based on virtual protection cycles," IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, Oct. 2000, pp. 1938-1949.

[23] [GrMaEL94] W. D. Grover, M. H. MacGregor, "Potential for spare capacity preconnection to reduce crossconnection workloads in mesh-restorable networks," Electronics Letters, vol. 30, no. 3, 3 Feb. 1994, pp. 194-195.

[24] [MaGrJNSM97] M. H. MacGregor, W. D. Grover, K. Ryhorchuk, "Optimal spare capacity preconfiguration for faster restoration of mesh networks," Journal of Network and Systems Management, vol. 5, no. 2, Jun. 1997, pp. 159-171.

[25] [ZhYaHSNMC02] Y.Zhang, O.Yang, "A distributed tree algorithm for WDM network for WDM network protection/restoration," Proceedings of the IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC02, Jeju Island, Korea, July 02, pp. 289-294

[26] [DuGrJSAC94] D.A. Dunn, W.D. Grover, M.H. MacGregor, "A comparison of k-shortest paths and maximum flow methods for network facility restoration," IEEE Journal on Selected Areas in Communications, Jan. 1994, vol. 12, no. 1, pp. 88-99.

[27] [KaGrGLOBECOM03] G. Kaigala, W.D. Grover, "On the Efficacy of GMPLS Auto-Reprovisioning as a Mesh-Network Restoration Mechanism," Proceedings of IEEE Globecom 2003, San Francisco, December 1-5, 2003, paper GC-13-6.

[28] [Bhan99] R. Bhandari, "Survivable Networks: Algorithms for Diverse Routing," Kluwer Academic Publishers, 1999.

[29] [IrMaTON98] R.R. Iraschko, M. MacGregor, W.D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh survivable networks," IEEE/ACM Transactions on Networking, vol.6, no.3, June 1998, pp. 325-336.

[30] [DoGrCCECE99] J. Doucette, W. D. Grover, R. Martens, "Modularity and Economy-of-Scale Effects in the Optimal Design of Mesh-Restorable Networks," Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 1999), Edmonton, AB, pp. 226-231, May 1999.

[31] [DoGrJSAC00] J. Doucette, W. D. Grover, "Influence of Modularity and Economy-of-scale Effects on Design of Mesh-Restorable DWDM Networks," IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Protocols and Architectures for Next Generation Optical WDM Networks, vol. 18, no. 10, pp. 1912-1923, October 2000.

[32] [GrDoDRCN01] W. D. Grover, J. Doucette, "Increasing the Efficiency of Span-Restorable Mesh Networks on Low-Connectivity Graphs," Proceedings of the 3rd International Workshop on Design of Reliable Communication Networks (DRCN 2001), Budapest, Hungary, pp. 99-106, October, 2001.

[33] [GrDoJSAC02] W. D. Grover, J. Doucette, "Design of a Meta-Mesh of Chain Sub-Networks: Enhancing the Attractiveness of Mesh-Restorable WDM Networking on Low Connectivity Graphs," IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on WDM-based Network Architectures, vol. 20, no. 1, pp. 47-61, January 2002.

[34] [DoGrOPTICOMM02] J.E. Doucette, W. D. Grover, "Capacity Design Studies of Span-Restorable Mesh Networks with Shared-Risk Link Group (SRLG) Effects," Proceedings of the SPIE Optical Networking and Communications Conference (OptiComm 2002), Boston, July 29-Aug. 2, 2002, paper 4874-3, pp.25-38.

[35] [BTF01] The New National Dream: Networking the Nation for Broadband Access, Report of the National Broadband Task Force, Government of Canada, June 18, 2001.

[36] [BELLLABS] Bell Labs Web-site, (http://www.bell-labs.com/project/feature/archives/glassceiling/) Online Reference: Accessed Jan 2004.

[37] [FCC1] FCC, office of Engineering and Technology Web-site, (http://www.fcc.gov/oet/). Online Reference: Accessed Jan 2004.

[38] [FCC2] FCC Outage Report 02-026 (http://www.fcc.gov/oet/). Online Reference: Accessed: Jan 2005

[39] [FCC3] FCC Outage Report 02-028 (http://www.fcc.gov/oet/). Online Reference: Accessed: Jan 2004.

[40] [FCC4] FCC Outage Report 02-034 (http://www.fcc.gov/oet/). Online Reference: Accessed: Jan 2004.

[41] [OPNET] OPNET Web-site, (http://www.opnet.com/products/modules/flow_analysis.html). Online Reference: Accessed: Jan 2004.

[42] [KiKoIETF01] S. Kini, M. Kodialam, T. V. Laksham, S. Sengupta, C. Villamizar, "Shared backup Label Switched Path restoration," IETF Internet Draft, draft-kini-restoration-shared-backup-01.txt, work in progress, May 2001.

[43] [MORLEY01] G. D. Morley, Analysis and Design of Ring-Based Transport Networks, University of Alberta Ph.D. Thesis, February 2001.

[44] [MaJSAC98] M. W. Maeda, "Management and Control of Transparent Optical Networks," IEEE Journal on Selected Areas in Communications (JSAC), vol. 16, no. 7, pp. 1005-1023, September 1998.

[45] [HeByTON95] M. Herzberg, S. J. Bye, A. Utano, "The Hop-Limit Approach for Spare-Capacity Assignment in Survivable Networks," IEEE/ACM Transactions on Networking, vol. 3, no. 6, pp. 775-784, December 1995.

[46] [XiMaTON99] Y. Xiong, L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," IEEE/ACM Transactions on Networking, vol. 7, no. 1, pp. 98-110, February 1999.

[47] [ScGrICC02]D. A. Schupke, C. G. Gruber, A. Autenrieth, "Optimal Configuration of p–Cycles in WDM Networks," Proceedings of IEEE International Conference on Communications (ICC 2002), New York, NY, pp. 2761-2765, April-May 2002.

[48] [KaHoNATURE01] J. M. Kahn, K.-P. Ho, "A Bottleneck for Optical Fibers," Nature, vol. 411, pp. 1007-1010, June 28, 2001.

[49] [NF01] http://www.nwfusion.com/edge/news/2001/123232_07-30-2001.html, Online Reference: Accessed Jan 2004.

[50] [KaIEEE00] S.V. Kartalopoulos, Introduction to DWDM Technology, IEEE Press and SPIE Press, 2000.

[51] [FREEMAN02] R.L. Freeman, Fiber-Optic Systems for Telecommunications, John Wiley & Sons, 2002. (See Chapters 6 and 10).

[52] [StGrTR99] D. Stamatelakis, W. D. Grover, "Network restorability design using pre-configured trees, cycles, and mixtures of pattern types," TRLabs, Edmonton, Alberta, Canada, Technical Report TR-1999-05, 30 Oct. 2000.

[53] [ShGrJSAC03] G. Shen, W. D. Grover, "Extending the *p*-cycle concept to path segment protection for span and node failure recovery," IEEE Journal on Selected Areas in Communications, vol. 21, no. 8, Oct. 2003, pp. 1306-1319.

[54] [ShYaPNET04] Shahram Shah-Heydari, Oliver Yang, "Hierarchical Protection Tree Scheme for Failure Recovery in Mesh Networks, " Photonic Network Communications, Volume 7, Issue 2, March 2004, Pages 145 – 159.

[55] [IrGrDRCN98] R. R. Iraschko, W.D. Grover, M.H. MacGregor, "A distributed real time path restoration protocol with performance close to centralized multi-commodity maxflow," Proc.1st Int'l Workshop on Design of Reliable Communication Networks, (DRCN'98), Brugge, Belgium, May 17-20, 1998, paper O.9

[56] [GrZhDRCN98] W. D. Grover, Y. Zheng, "VP-based ATM network design with controlled over-subscription of restoration capacity," Proc.1st Int'l Workshop on Design of Reliable Communication Networks (DRCN'98), Brugge, Belgium, May 17-20, 1998, paper O.33.

[57] [DoClONM03] J. Doucette, M. Clouqueur, W. D. Grover, "On the availability and capacity requirements of shared backup path-protected mesh networks," Optical Networks Magazine, Nov./Dec. 2003, pp. 29-44.

[58] [OuICC00] C. Ou et al., Near-Optimal Approaches for Shared-Path Protection in WDM Mesh Networks," IEEE International Conference on Communications, May, 2003, Page(s): 1320—1324.

[59] [XiXuJLWT03] Y. Xiong, D. Xu and C. Qiao, Achieving Fast and Bandwidth Efficient Shared-path Protection, IEEE Journal of Lightwave Technology, Vol. 21, No. 2, Feb. 2003, pp. 365-371.

[60] [StMSC97] Demetrios Stamatelakis, "Theory and Algorithms for Preconfiguration of Spare Capacity in Mesh Restorable Networks," M.Sc. Thesis submitted to the University of Alberta, Spring 1997.

[61] [BaCOST239] P. Batchelor et al., "Ultra high-capacity optical transmission networks: Final report of action COST 239," Faculty of Electrical Engineering and Computing, University of Zagreb, 1999.

[62] [GrClICOCN02] W. D. Grover, M. Clouqueur, "Span-Restorable Mesh Network Design to Support Multiple Quality of Protection (QoP) Service-Classes," Conf. on Optical Communications and Networks (ICOCN'02), Singapore, Nov.11-14, 2002, pp. 321-323

[63] [ClGrDRCN01] M. Clouqueur, W. D. Grover, D. Leung, O. Shai, "Mining the rings: strategies for ring-to-mesh evolution," Proc. Design of Reliable Communication Networks (DRCN 2001), Budapest, Hungary, October 2001, pp.113-120.

[64] [RaMuINFOCOM99] S. Ramamurthy, B. Mukherjee, "Survivable WDM mesh networks, Part I – Protection," Proc. 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99), New York, USA, 21-25 Mar. 1999, vol. 2, pp. 744-751.

[65] [GrIEEE97] W. D. Grover, "Self-organizing broadband transport networks," Proceedings of the IEEE, vol. 85, no. 10, Oct. 1997, pp. 1582-1611.

[66] [GrIEEE94] W. D. Grover, "Distributed restoration of the transport network," in Telecommunications Network Management into the 21st Century, S. Aidarous and T. Plevyak, Eds. New York: IEEE Press, 1994, pp. 337-419.

[67] [ScGrICC04] D. A. Schupke, W. D. Grover, M. Clouqueur, "Strategies for enhanced dual failure restorability with static or reconfigurable *p*-cycle networks," Proc. IEEE International Conference on Communications (ICC) 2004, Paris, France, 20-24 Jun. 2004.

[68] [ShGrOPTICOMM03] G. Shen, W. D. Grover, "Exploiting forcer structure to serve uncertain demands and minimize redundancy of *p*-cycle networks," Proc. Fourth International Conference on Optical Networking and Communications (OptiComm 2003), Dallas, Texas, USA, 13-17 Oct. 2003.

[69] [AN95] ANSI, Synchronous Optical Network (SONET) – Automatic Protection Switching, ANSI T1.105.01-1995.

[70] [AN95a] ANSI, Synchronous Optical Network (SONET) – Basic Description including Multiplex Structure, Rates and Formats, ANSI T1.105-1995, 1995.

[71] [BELL95] Bellcore, SONET Dual-Fed Unidirectional Path Switched Ring (UPSR) Equipment Generic Criteria, GR-1400-Core, Issue 1, Revision 1, Oct 1995

[72] [BELL95a] Bellcore, SONET Bidirectional Line Switched Ring Equipment Generic Criteria, GR-1230-Core, Issue 2, Nov 1995.

[73] [NeHaOFC99] P. Neusy, R. Habel, "Availability Analysis of Optical Shared Protection Rings for Long-haul Networks," Proceedings of Optical Fiber Communication Conference (OFC 1999), San Diego, CA, pp. TuL5-1–TuL5-4, February 1999.

[74] [CISC99] Cisco Systems, Dynamic Packet Transport Technology, A Cisco Systems White Paper, 1999.

[75] [BoRo02] P.Bonefant, A. Rodriguez-Moral, "Generic Framing Procedure (GFP):The catalyst for Efficient Data over Transport," IEEE Communication Magazine, May 2002, pp 72-79.

[76] [CoMa02] IEEE Communications Magazine, Special Issue on GFP and DOS and OTN, vol 40, no 5, may 02, pp 60-112.

[77] [ClGrPNET03] M. Clouqueur, W.D. Grover, "Mesh-restorable networks with enhanced dual-failure restorability properties," Photonic Network Communications, Springer Science, 9:1, 2005, pp. 7-18.

[78] [KoGrJLT] A. Kodian, W.D. Grover, "Failure independent path-protecting *p*-cycles: efficient and simple fully pre-connected optical-path protection," *IEEE Journal of Lightwave Technology*, vol. 23, no.10, October 2005.

[79] [GrKoUSPT] W.D.Grover, A.Kodian, "Failure-Independent Path Protecting *p*-Cycles", provisional US Patent, filed 29th Nov, 2004.

[80] [KoSaOSN] A. Kodian A. Sack, Wayne D. Grover, "The threshold hop-limit effect in *p*-cycles: Comparing hop- and circumference-limited design," Optical Switching and Networking, Elsevier, July 2005, pp. 72-85.

[81] [KoSaBROADNETS04] A. Kodian, A. Sack, W. D. Grover, "*p*-Cycle Network Design with Hop Limits and Circumference Limits," Proceedings of the First International Conference on Broadband Networks (BROADNETS 2004) , San José, California, USA, 25-29 October 2004, pp. 244-253

[82] [KoGrDRCN05] A. Kodian, W. D. Grover, J. Doucette, "A Disjoint Route Sets Approach to Design of Failure-Independent Path-Protecting *p*-Cycle Networks," in 5th International Workshop on Design of Reliable Communication Networks (DRCN 2005), Ischia (Naples), Italy, 16-19 October 2005.

[83] [ART03] Executive summary of a report to the ART, (January 2003) "Developments in International Telecom," Available: Dec 2003.

[84] [GeSaOPTICOMM01] O. Gerstel, G. Sasaki, "Quality of Protection (QoP): A Quantitative Unifying Paradigm to Protection Service Grades," Proc. OptiComm2001, Aug. 2001.

[85] [SPIE01] S. Kartalopoulos, "What is WDM," SPIE OER report . Online publication Available Feb 05, 2004: http://www.spie.org/web/oer/november/nov00/wdm.html

[86] [CaOR02] E.Castillo et.al, "Building and solving mathematical programming models in engineering and science," textbook by John Wiley & Sons, Edition: 2002.

[87] [Lt01] Online reference: Accessed March18th, 2005.
Available at http://www.lightreading.com/document.asp?doc_id=995&print=false

[88] [WIRED01] http://www.wirelessweek.com/article/CA3588?spacedesc=&stt=001, Online Reference: Available Jan 2006.

[89] [IT01] http://www.itworldcanada.com/Pages/Docbase/ViewArticle.aspx?ID=idgml-47b7fa7b-e8f-4619-9438-468224f2c3df, Online Reference: Available Jan 2006.

[90] [BiKeONM03] G. Birkan, J. Kennington, E. Olinick, A. Ortynski, G. Spiride. 2002. Making a Case for Using Integer Programming to Design DWDM Networks. Optical Networks Magazine. 4, 6 (2003) 107-120.

[91] [GrDoAOR01] W.D. Grover, J. Doucette, "Topological design of survivable mesh-based transport networks," Annals of Operations Research: Topological Network Design in Telecommunication Systems, P. Kubat, J. MacGregor Smith (Editors), P.L. Hammer (Editor-in-Chief), Kluwer Academic Pub., vol. 106 (2001), September 2001, pp.79-125.

[92] [Feyn02] R.L. Feynman, "Fiber-optic systems for telecommunications," Wiley series in telecommunications and signal processing, J.G.Proakis editor.

[93] [NTT01] http://www.physorg.com/news3316.html, Online Reference: Available Jan 2006.

[94] [StGrUS02] D. Stamatelakis, W. D. Grover, "Scalable network restoration device," U.S. Patent 6,404,734, 11 Jun. 2002.

[95] [StGrCAN02] D. Stamatelakis, W. D. Grover, "Scalable network restoration device," Canadian Patent Application 2280981, 27 Aug. 1999.

[96] [GrStUS02] W. D. Grover, D. Stamatelakis, "Protection of routers in a telecommunications network," U.S. Patent Application, 26 Apr. 1999.

[97] [GrStCAN99] W. D. Grover, D. Stamatelakis, "Protection of routers in a telecommunications network," Canadian Patent Application 2269649, 22 Apr. 1999.

[98] [ShGrCAN03]G. Shen, W. D. Grover, "Path segment protecting *p*-cycles," Canadian Patent Application 2435193, 14 Jul. 2003.

[99] [GrShUS03] W. D. Grover, G. Shen, "Path segment protecting *p*-cycles," U.S. Patent Application 20040109407, 14 Jul. 2003.

[100] [GrDoUS03] W. D. Grover, J. Doucette, "Method for design of networks based on *p*-cycles," U.S. Patent Application 20040133663, 2 Jul. 2003.

[101] [GrDoCAN03] W. D. Grover, J. Doucette, "Method for design of networks based on *p*-cycles," Canadian Patent Application 2434115, 30 Jun. 2003.

[102] [GrUS02] W. D. Grover, "Distributed preconfiguration of spare capacity in closed paths for network restoration," U.S. Patent 6,421,349, 16 Jul. 2002.

[103][StGrUS02a] D. Stamatelakis, W. D. Grover, "Distributed preconfiguration of spare capacity in closed paths for network restoration," U.S. Patent Application 20030048749, 2 Jan. 2002.

[104][Faber01] P. Ormerod, "Why Most Things Fail, *Faber and Faber* 2005

[105][ToNe94] M. To, P. Neusy, "Unavailability Analysis of long-haul networks," *IEEE JSAC.*, vol. 12, no. 1, Jan 1994, pp 100-109.

[106][Tham01] I. Tham, ZDNet UK, (20 Sep. 2001) [Online], "Anchor-draggers cut Asia's Internet pipe," Available: http://news.zdnet.co.uk/Internet/0,39020369,2095715,00.htm.

[107][BoBr98] L. Bowman, M. Broersma, ZDNet News, (10 Jun. 1998) [Online], "Severed MCI cable cripples the Net," Available: http://zdnet.com.com/2100-11-510740.html, Accessed: 20 Jan. 2004.

[108][We03] G. Wearden, ZDNet UK, (26 Nov. 2003) [Online], "Cable failure hits UK Internet traffic," http://news.zdnet.co.uk/communications/networks/0,39020345,39118125,00.htm. Available Jan 2005.

[109][Mc01] W. McAuliffe, ZDNet UK, (3 Aug. 2001) [Online], "Train crash could be to blame for Internet derailment," Available: http://news.zdnet.co.uk/business/0,39020645,2092503,00.htm.

[110][QSN03] Quebec Scientific Information Network, (29 Jan. 2003) [Online], "Forestville-Rimouski underwater cable repaired," Available: http://www.risq.qc.ca/nouvelles/nouvelle_item.php?LANG=EN&ART=1231.

[111][McDonald94] J. C. McDonald, "Public network integrity – Avoiding a crisis in trust," IEEE Journal on Selected Areas in Communications, vol. 12, no. 1, Jan. 1994, pp. 5-12.

[112][Cl04] M. Clouqueur, "Availability of service in mesh-restorable transport networks," Ph.D. Dissertation, University of Alberta, Edmonton, Alberta, Canada, Spring 2004.

[113][Craw93] D. Crawford "Fiber Optic Cable Dig-ups," in Network Reliability: A Report to the National – Compendium of Technical Papers, National Engineering Consortium, Chicago, June 1993

[114][DaReKauffman00] B.Davie, Y. Rekhter, MPLS: Technology and Applications, Morgan Kaufmann, CA, USA 2000.

[115][FoMcGraw98] B. Forouzan, Introduction to Data Communications and Networking, Boston, Massachusetts, McGraw-Hill, 1998.

[116][RaKauffman02] R. Ramaswami, K.N. Sivarajan, Optical Networks: A Practical perspective, Kaufmann Publishers, CA, 2002.

[117][StBa00] T. E. Stern, K. Bala, Multiwavelength Optical Networks: A Layered Approach, Prentice Hall, Upper Saddle River, NJ, 2000.

[118][ToSc02] P. Tomsu, C. Schmutzer, Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technologies, Prentice Hall, Upper Saddle River, NJ, 2002.

[119][Dixi03] , "IP over WDM: Building the next Generation Optical Internet," *John Wiley and Sons*, 2003.

[120][Rasa97] R.Ramaswami, G.H. Sasaki. "Multi-wavelength optical networks with limited All-Optical Networks," *IEEE TON*, Vol 3, 1995, pp. 489-500.

[121][ShBo00] G.Shen, S.K. Bose et al., "Efficient wavelength assignments for light-paths in WDM optical networks with/without wavelength conversion," *Photonic Network Communications*, vol. 2, pp. 3490360, Nov 2000.

[122][CiscONS][Online] http://www.cisco.com/en/US/products/hw/modules/ps5791/ps5814/, Accessed: Dec 2005

[123][Brun00]D. A. Brungard, "Draft ITU-T G.70-9 Network Node Interface for the OTN," *Contribution to T1 Standards, Project T1X1.5/2000-091*, Feb. 2000.

[124][802.3ae] [Online] http://grouper.ieee.org/groups/802/3/ae/public/jul01/bynum_1_0701.pdf , Accessed: Dec 2005.

[125][ToRaICCN04] A. Todimala, B. Ramamurthy, "IMSA: An Algorithm for SRLG Diverse Routing in WDM Mesh Networks," *Proc. ICCCN*, Oct 2004.

[126][LiLiDRCN05] *p*-Cycle Design in Survivable Networks with Shared Risk Link Groups (SRLGs)Chang Liu, Lu Ruan - Iowa State University – DRCN 2005

[127][KiLu04] S. Kim, S. Lumetta, "Capacity-Efficient Protection with Fast Recovery in Optically Transparent Mesh Networks," Proceedings of the First International Conference on Broadband Networks (BROADNETS 2004) , San José, California, USA, 25-29 October 2004.

[128][KoZyJNSM05] A. Koster, A.Zymolka, M.Jaeger, R. Huelsermann, "Demand-wise shared protection for meshed optical networks," *JNSM*, vol. 13, No. 1, 2005.

[129][KoZyTech04] A. Koster et.al., [ONLINE] http://www.zib.de/Publications/Reports/ZR-03-24.pdf, Accessed: Dec 2005.

[130][ZyKoONDM03] A. Zymolka, A. Koster, and R. WessÄaly. Transparent optical network design with sparse wavelength conversion. In Proc. ONDM 2003, Budapest, Hungary, February 2003.

[131][KoDRCN03] A. Koster et al, "Demand-wise Shared Protection for Meshed Optical Networks," Proc. Design of Reliable Communication Networks, Banff, Alberta, Canada, Oct 19-22, 2003. pp. 85-92.

[132][KuTh04] E. Kublinskas, "Designing Resilient and Fair Multi-layer Telecommunication Networks," Licentiate Thesis, Department of Communication Systems, Lund University, Sweden, 19 February, 2005

[133][GrMa94] W.D. Grover and M.H. MacGregor, "On the potential for spare capacity preconnection to reduce cross connection workloads in mesh-restorable networks," Electronics Letters, Feb. 3, 1994, vol. 30, no.3, pp. 194-195.

[134][Grov97] W.D. Grover, "Self-organizing broadband transport networks," Proceedings of IEEE, Special Issue on Communications in the 21st Century. vol.85, no.10, October 1997, pp.1582-1611.

[135][Krey] E. Kreyszig, " Advanced Engineering Mathematics," John Wiley and Sons, Fourth Edition, 2004.

[136][Winst] W. Winston, "Operations Research Applications and Algorithm," Duxbury Press, Belmont, CA, 1994.

[137][Dantzig] G. Dantzig, Linear Programming and Extensions, Princeton, New Jersey, Princeton University Press, 1963.

[138][CPLEX] ILOG, "ILOG CPEX," accessed Dec 2005, www.ilog.com/products/cplex, 2005.

[139][AMPL] R. Fourer, D. M. Gay, B. W. Kernighan, AMPL: A Modeling Language For Mathematical Programming, 2nd edition, Pacific Grove, California, Brooks/Cole – Thompson Learning, 2003.

[140][CP-OR] Operations Research, "Online http://clip.dia.fi.upm.es/~vocal/public_info/seminar_notes/node9.html

[141][Knuth] D. Knuth, "The Art of Computer Programming, Vol 1, Fundamental Algorithms," Third Edition, Addison-Wesley, 1997, pp. 107-123.

[142][CoLe01] T. Cormen et al. "Introduction to Algorithms," Second Edition, McGraw-Hill, 2001,pp. 41-50.

[143][CrVi98] P. Crescenzi, V. Kann, "How to find the best approximation results," A compendium of NP optimization problems, *ACM SIGACT News,* Dec 1998.

[144][Dijk] E. Dijkstra, "A note on two problems in connection with graphs," *Number Math.*, vol. 37, No. 8, 1999, pp. 70-75.

[145][SeWeNFOEC02] M. Severi, G. Wellbrock, "Migration from point-to-point and ring architectures to meshed networks in DWDM: limitations and possible architectures," Proc. National Fiber Optics Engineers Conference 2002, pp 2010.

[146][JoMu03] J. John, T Mueller, [Online] "From Ring to Mesh: Why, When and How?" Available: http://www.lucent.com/livelink/090094038000f714_White_paper.pdf. [May 18th 2003].

[147][Abowd03] M. Abowd, [Online] "Ring and mesh networking, a peaceful coexistence," Available: http://www.lucent.com/livelink/0900940380004b81_White_paper.pdf. [May 18th 2003].

[148][GrClLe02] W. Grover, M. Clouqueur, D. Leung, "Evolution of a Telecommunication Network from a Ring to Mesh Structure," U.S. Patent Pending, filed June 2002.

[149][LeGrMo99] C-Y. Lee, W. D. Grover, G. D. Morley, "Heuristic methods for the span elimination problem in ring-based transport network design," IEEE CCECE'99, Edmonton, Canada, May 1999.

[150][DoGrOFC01] J. Doucette, W. D. Grover, T. Bach, "Bi-criteria studies of mesh network restoration path-length versus capacity tradeoffs," Proc. Optical Fiber Communication Conference (OFC) 2001, Anaheim, California, USA, 17-22 Mar. 2001, pp. TuG2-1-TuG2-3.

[151][RyPaICOIN] M. S. Ryu, H. S. Park, "Survivable network design using restricted *p*-cycle," Proc. International Conference on Information Networking (ICOIN) 2003, Jeju Island, Korea, 12-14 Feb. 2003.

[152][KoGrBROADNETS05] A. Kodian, W.D. Grover, "Multiple-Quality of Protection Classes Including Dual-Failure Survivable Services in *p*-Cycle Networks," *Proc. IEEE BROADNETS 2005*, Boston, MA, USA.

[153][GrKoICTON] W.D. Grover, A. Kodian, "Failure-Independent Path Protection with *p*-Cycles: Efficient, Fast and Simple Protection for Transparent Optical Networks," in Proc. 7th Int'l Conf. on Transparent Optical Networks (ICTON 2005), July 3-7, Barcelona, Spain, (6ms)

[154][KoBaPNET06] A.Kodian, D. Baloukov, W.D. Grover, "Joint Design of Failure Independent Path Protecting *p*-Cycle Networks," submitted to *Kluwer Photonic Network Communications*, 2006.

[155][VeHa96] P. Veitch, I. Hawker, and G. Smith, "Administration of restorable vir-tual path mesh networks," IEEE Communications Magazine, Vol 34, No 12, December 1996, pp. 96-101.

[156][Maxim] Online Reference: http://pdfserv.maxim-ic.com/en/an/3258.pdf Accessed Jan 10, 2006.

[157][Keiser] G. Keiser, "Optical Fiber Communications," McGraw Hill International Edition, 2000.

[158][Palais] J.Palais, "Fiber Optic Communications", Fourth Edition Prentice Hall, Upper Saddle River, NJ, 1998

[159][Hecht] J. Hecht, "Understanding Fiber Optics, Third Edition," Prentice Hall, Upper Saddle River, NJ, 1999.

[160][InteroperabilityTutorial] Online Reference: http://www.fcc.gov/oet/tutorial/9-24-04-oni-tutorial.ppt Accessed Jan 10, 2006.

[161][ScICC03] Schupke, D.A., "The Tradeoff Between the Number of Deployed *p*-Cycles and the Survivability to Dual Fiber Duct Failures," IEEE International Conference on Communications (ICC), Anchorage, AK, USA, May 11-15, 2003.

[162][GrPNET05] A. Groebbens et al, "Efficient Protection in MPλS networks Using Backup Trees: Part Two- Simulations," Photonic Network Communications, Kluwer, Vol 6(3), pp. 207-222.

[163][GrPNET05b]A. Groebbens et al, "Efficient Protection in MPλS networks Using Backup Trees: Part One- Concepts and Heuristics," *Photonic Network Communications*, Kluwer, Vol 6(3), pp. 191-206.

[164][KoGrOSN06] A.Kodian, W.D. Grover, "Multiple Quality of Protection in *p*-Cycle Networks," submitted to *Elsevier Journal of Optical Switching and Networking*, Expected Fall. 2006.

[165][DoGeDRCN05] J. Doucette, P. Giese, W. D. Grover, "Combined Node and Span Protection Strategies with Node-Encircling *p*-Cycles," in 5th International Workshop on Design of Reliable Communication Networks (DRCN 2005), Ischia (Naples), Italy, 16-19 October 2005.

[166][Lau89] H.T. Lau, "Algorithms on Graphs," *TAB Professional Reference Book*, TAB Books Inc., PA, 1989.

[167][BaJe76] Barnoski M.K., Jensen S.N., "Fiber Waveguides: a Novel Technique for Investigating Attenuation Characteristics", *Applied Optics 15(9), (1976)* 2112-2115.

[168] [KeWired] K. Poulsen, “The Backhoe: A Real Cyberthreat,” *Article in WIRED news*, Online Reference: http://wired.com/news/technology/0,70040-0.html?tw=wn_tophead_1, Accessed: Jan 28th, 2005.

[169] [GrJLT88] W.D. Grover, "Forward Error Correction in Dispersion-Limited Lightwave Systems," *IEEE Journal of Lightwave Technology*, vol. 6, no. 5, May 1988, pp. 643-654

Footnotes

---

[i] Traffic Grooming is a separate area that studies the efficient packing of multiple lower data rates (33.6kbs modems, DSL etc.) into one larger data rate backbone. This is somewhat analogous to when you fly from Edmonton to Tokyo. A smaller plane (smaller data rate) takes a few people from Edmonton to Denver. These people are then aggregated with other Tokyo-bound passengers from numerous other smaller points of origin on a large Jumbo jet. This achieves better fuel efficiency on the long-haul portion by achieving a better matching of plane size to the traffic on each leg of the route.

[ii] On a related note the CRTC, the Canadian telecommunications regulation agency recently mandated that telephony customers may also be reimbursed for poor quality service and outages.

iii Graph theory is sometimes used for illustrative purposes and to abstract network information. A span in the network is usually represented as an edge in the network graph and a node in the network is represented by a vertice in the network graph. Thus the popular notation for a network graph is G = (V,E) where V is a set of nodes and E is the set of spans. In subsequent discussions we will use the word span and node to also refer to edges and vertices, respectively, in the corresponding network graph.

[iv] As an aside, it has been shown that full node failure survivability can be achieved under the capacity provisioned for single span failure survivability, primarily because in the event of a node failure, only transiting demands can survive. This takes a significant load off the survivability mechanism as the demands that originate/terminate at the node do not use the backup/working capacity. Thus nodes have to be physically protected if the local demands are to survive.

[v] Max flow refers to one of two possible variations on a commonly found design problem: a) calculate the maximum flow between two end-nodes under existing capacity assignments on all edges or b) calculate the minimum network capacity required to support a give number of flow units between end-node pairs. See [DuGrJSAC94] and pp. 236 – 244 of [Grov03] for more information.

[vi] Mutual capacity is a problem specific to any attempt at end-to-end path restoration. A single span failure may simultaneously affect multiple end-to-end working paths. Each affected working path requires restoration. One working path may have 20 possible backup options whereas another working path may have only one. How does one fundamentally ensure that the first backup path chosen does not block the only backup path available to another demand that has also failed? For more details on mutual capacity, we refer you to [Grov03] and [IrMaTON98].

[vii] Fault localization refers to the determination of which span or node actually was the cause of the path failure. This information is needed as part of a failure-specific response, such as for path restoration [IrMaTON98]. In a transparent optical network this information is, however, not necessarily available quickly because the payload is not accessed electronically at every intermediate node to inspect its overhead bytes. In addition, transparent optical cross-connects typically do not generate "keep alive" optical output signals, so that when an outright loss of light occurs, this alarm goes off at all downstream nodes. A centralized interrogation of alarm data may then be needed to find the node nearest to the actual physical source of the failure. Failure independence of the protection reaction is therefore desirable as it requires only end-node failure detection, not failure localization, to be able to act.

[viii] Each additional hop in the path implies additional signalling to seize and cross-connect the respective shared spare channel, including checks that it is not already in use. A pre-connected protection path has no such delays. The only delays are the time to switch the failed signal path over into the protection structure at the end-nodes, followed by physical propagation and reframing times. The latter are common to any possible protection scheme other than dedicated 1+1 APS, however.

[ix] Standard capacity redundancy is the ratio of total protection capacity-distance plus extra working capacity-distance over shortest paths to the working capacity needed only for a shortest-path realization of the working paths alone. This avoids the degeneracy of measuring redundancy when working paths are themselves made longer than shortest paths. Otherwise, by increasing working path lengths one can apparently reduce redundancy without reducing absolute spare capacity. (See [Grov03] pp. 47-49)

[x] It might be pointed out that in [KoZyTech04] larger savings, up to 28% relative to 1+1 APS, are achieved if only 2/3rds of the network demands are to be protected. But simple omission of any desired fraction of total demand from the protected demand is also an option in any scheme to reduce overall protection costs.

[xi] Triple and quadruple span failures are not considered because it is very rare for three or four spans to fail simultaneously. In addition, so many simultaneous failures can rarely be remedied by design alone. In the studies in this thesis, we consider only single and dual span failure scenarios

.

[xii] This is extremely rare and is called Total Unimodularity. A TU ILP is one that produces an integral solution when all variables are LP relaxed. An example of a TU ILP is the transportation problem in 3.5.1. The reader will note that none of the variables were forced to be integers but the resultant solution is integral. Currently, all analytical methods to verify whether an ILP is TU are also fairly involved. In practice, the quickest way to check is to run the LP relaxed version and examine the results to see if all variables are indeed integral.

[xiii] Optical signal reach is defined as the maximum distance a WDM signal can travel before it requires 3R regeneration. Equipment designers are always trying to improve optical reach as it reduces the number of expensive 3R regenerators that have to be placed in the network.

[xiv] Notably for this writer, the first ever proposal and technical analysis for the case for FEC in fiber optic transmission systems was coincidentally made by the supervisor of this work in 1988 [GrJLT88
]

[xv] Any other pattern of non-uniform growth multipliers can be used in an actual planning study using these methods.

[xvi] Bi-criteria objective functions as applied to network design are studied in [DoGrOFC01]. The logic behind bi-criteria objective functions comes from a fundamental property of ILPs. In most cases, in a given solution space, many different optimal solutions exist. While it is guaranteed that all optimal solutions will have the same value of objective function, they all differ in the details. In the current context, adding a bi-criteria objective is like asking the solver, "if there are multiple optimal solutions, can you return one that also uses shorter path lengths, on average?" A quick way of doing this is to simply add a weighted variable term in the objective function. The weight, ε, is usually a very small value (0.0001 in our case) and ensures that the value of the second term in the objective function doesn't actually significantly change the objective function.

[xvii] Multiple quality of service classes are well understood by another transport industry, the airlines. Most flights may have multiple service classes on board–Executive, First, Enhanced Economy, Economy/Coach–but also tiers of price classes based on booking time. The ultimate aim to maximize revenue through responsiveness to the different actual needs of customers. The aim here is to develop a customer-need responsive optical network.

[xviii] Readers may find it interesting that we first describe the straddling span routing principle for dual failure survivability, and subsequently exploit it in the design model. But understanding of this principle actually came in the reverse sequence: First a general model was created that presupposed nothing about routing choices whatsoever, and the dual failure survivability requirements simply asserted on some demands. Upon inspecting the results, we found that the p-cycles had been chosen such that all routes for platinum services only ever crossed straddling spans! In hindsight, this explained as above. It is an example of how networking science can benefit from ILP models as research tools to discover such insights.

[xix] Standard capacity redundancy is the ratio of total protection capacity-distance plus extra working capacity-distance over shortest paths to the working capacity needed only for a shortest-path realization of the working paths alone. This avoids the degeneracy of measuring redundancy when working paths are

---

themselves made longer than shortest paths. Otherwise, by increasing working path lengths one can apparently reduce redundancy without reducing absolute spare capacity. (See [Grov03] pp. 47-49)

[xx] Each additional hop in the path implies additional signaling to seize and cross-connect the respective shared spare channel, including checks that it is not already in use. A pre-connected protection path has no such delays. The only delays are the time to switch the failed signal path over into the protection structure at the end-nodes, followed by physical propagation and reframing times. The latter are common to any possible protection scheme other than dedicated 1+1 APS, however.

[xxi] This is a reasonable assumption in joint optimal designs as in [GrDoLEOS02] because the solver will be strongly biased towards producing relationships where straddlers are can fully utilize both sides of the FIPP *p*-cycle.

xxii Shortest path working routing may not always result in a feasible FIPP *p*-cycle solution in a network that has traps. For a comprehensive examination of trap-avoiding routing strategies see [Bhan99]. We deal with traps using Bhandari's digraph based solution to Suurballe's problem also in [Bhan99].

Appendix 1. AMPL Models

1. Ring Mining

```
#####################################################################################
#          PARAMETERS
#####################################################################################
# SETS


set SPANS;
# Set of all spans.

set PCYCLES;
# Set of all p-cycles.

set DEMANDS;
# Set of all O-D demand pairs.

set WORK_ROUTES{r in DEMANDS};
# Set of all working routes for each demand pair r.

#####################################################################################

param Cost{j in SPANS};
# Cost of each unit of capacity on span j.

param Work{j in SPANS};
# Number of working links placed on span j.

param Xpi{p in PCYCLES, i in SPANS} default 0;
# Number of paths a single copy of p-cycle p provides for restoration of failure of
# span i (2 if straddling span, 1 if on-cycle span, 0 otherwise).

param pCrossesj{p in PCYCLES, j in SPANS} := sum{i in SPANS: i = j and Xpi[p,j] = 1} 1;
# Equal to 1 if p-cycle p passes over span j, 0 otherwise.
# i.e. if Xpi[p,j] = 1, then p-cycle p crosses span j.

param DemandUnits{r in DEMANDS} default 0;
# Number of demand units between node pair r.

param ZetaWorkRoute{j in SPANS, r in DEMANDS, q in WORK_ROUTES[r]} default 0;
# Equal to 1 if qth working route for demand between node pair r uses span j and 0 otherwise.

param RingCapacity{i in SPANS};

#####################################################################################
#          VARIABLES
#####################################################################################

var workflow{r in DEMANDS, q in WORK_ROUTES[r]} >=0, <=10000;
# Working capacity required by qth working route for demand between node pair r.

var work{j in SPANS} >=0, <=10000 integer;
# Number of working wavelengths placed on span j.

var p_cycle_usage{p in PCYCLES} >=0 integer, <=10000;
# Number of copies of p-cycle p used.

var spare{j in SPANS} >=0 integer, <=10000;
# Number of spare links placed on span j.

var total_cost_spare >=0, <=1000000000000;
# Total cost of spare capacity.
```

```
var total_cost_work >=0, <=1000000000000;
# Total cost of working capacity.

var lambda >=1;


################################################################################
#          OBJECTIVE FUNCTION
################################################################################

#minimize TotalCost: total_cost_spare + total_cost_work;
minimize mult: 0-lambda;

# Minimize the total cost of capacity.
# Total costs of working and spare are calculated individually below as variables.
# We do it this way so that we can simply look at the values of those two variables
# to determine the separate costs of working and spare (instead of needing to set up
# a spreadsheet with individual span capacities to calculate them).


################################################################################
#              CONSTRAINTS
################################################################################

 subject to demands_met{r in DEMANDS}:
 sum{q in WORK_ROUTES[r]} workflow[r,q] >= lambda * DemandUnits[r];
 # All demands must be fully routed.

 subject to working_capacity_assignment{j in SPANS}:
 work[j] = sum{r in DEMANDS, q in WORK_ROUTES[r]: ZetaWorkRoute[j,r,q]=1} workflow[r,q];
 # There must be enough working capacity on span j to accomodate all working flows
 # simultaneously routed over it by all demand pairs.

 subject to full_restoration{i in SPANS}:
 work[i] <= sum{p in PCYCLES} Xpi[p,i] * p_cycle_usage[p];
 # Enough p-cycles must be placed so that each span failure is fully restorable.

 subject to spare_capacity_placement{j in SPANS}:
 spare[j] = sum{p in PCYCLES} pCrossesj[p,j] * p_cycle_usage[p];
 # Enough spare capacity is placed on each span to accomodate all the p-cycles
 # placed on it.

 subject to calulate_spare_cost:
 total_cost_spare = sum{j in SPANS} Cost[j] * spare[j];
 # The total cost of spare capacity is the sum of the costs of spare on each span.

 subject to calulate_work_cost:
 total_cost_work = sum{j in SPANS} Cost[j] * work[j];
 # The total cost of working capacity is the sum of the costs of working on each span.

 subject to ringcap{i in SPANS}:
 work[i] + spare[i] <= RingCapacity[i];
 # Total working and spare allocation on each span must not exceed the size of the
 # original ring span.
```

2. *p*-Cycle Design with Hop and Circumference Limits:

```
################################################################################
# ABSTRACT
#
# This model is used to design modular JCP p-cycle networks with path length
# restrictions. (Note: the convention we use is that the shorter side of the
# p-cycle is always L, and the longer side is always R.
################################################################################
```

```
################################################################################
# SETS
################################################################################

set MODULE_TYPES;
# Set of available module types, indexed by m.

set SPANS;
# Set of spans, indexed by i (failed) or j (surviving).

set DEMANDS;
# Set of demand relations, indexed by r.

set CYCLES;
# Set of eligible cycles, indexed by p.

set WORK_ROUTES{r in DEMANDS};
# Set of eligible working routes for each demand relation r, indexed by q.

set ALLWORKROUTES;

################################################################################
# PARAMETERS
################################################################################

param Large default 100000;
# A large positive constant (100000).

param ModuleCost{m in MODULE_TYPES};
# Cost of one module of type m.

param SpanCost{j in SPANS};
# Cost of span j (includes length, and other costs).

param ModuleSize{m in MODULE_TYPES};
# Capacity in STS-1 equivalents of module type m.

param DemandUnits{r in DEMANDS} default 0;
# Number of demand units in STS-1 equivalents for relation r.

param ZetaWorkRoute{j in SPANS, r in DEMANDS, q in WORK_ROUTES[r]} default 0;
# Equal to 1 if route q is through span j for relation r, 0 otherwise.

param Xpi{p in CYCLES, i in SPANS} default 0;
# Equal to 1 if p-cycle p can provide an acceptable restoration
# path for failure of span i, 0 otherwise (or can be the actual hop value) - not used.

param Xpi_L{p in CYCLES, i in SPANS} default 0;
# Equal to 1 if the L side of p-cycle p can provide an acceptable restoration
# path for failure of span i, 0 otherwise (or can be the actual hop value).

param Xpi_R{p in CYCLES, i in SPANS} default 0;
# Equal to 1 if the R side of p-cycle p can provide an acceptable restoration
# path for failure of span i, 0 otherwise (or can be the actual hop value).

param CYCLE_HOPS{p in CYCLES} default 0;
# This is the number of hops for each cycle - not used.

param pCrossesj{p in CYCLES, j in SPANS} default 0;
# Equal to 1 if cycle p lies on span j, 0 otherwise.

param SpanWiseElligibility_L{p in CYCLES, i in SPANS} default 0;
param SpanWiseElligibility_R{p in CYCLES, i in SPANS} default 0;
#dummy parameters to keep ampl happy - used to debug dat files.

param WorkHops{q in ALLWORKROUTES};

################################################################################
```

```
# VARIABLES
##########################################################################################

var number_modules{j in SPANS, m in MODULE_TYPES} >= 0 <= Large integer;
# Number of modules of type m placed on span j.

var work{j in SPANS} >=0 <= Large integer;
# Working capacity on span j.

var spare{j in SPANS} >= 0 <= Large integer;
# Spare capacity on span j.

var workflow{r in DEMANDS, q in WORK_ROUTES[r]} >= 0 <= Large integer;
# Number of units for relation r that use route q.

var p_cycle_usage{p in CYCLES} >= 0 <= Large integer;
# Number of unit-capacity copies of cycle p.

var p_cycle_usage_span_rest{i in SPANS, p in CYCLES} >= 0 <= Large integer;
# Number of unit-capacity copies of cycle p required for restoration of span i.

var p_cycle_usage_span_rest_l{i in SPANS, p in CYCLES} >= 0 <= Large integer;
# Number of unit-capacity copies of cycle p required for restoration of span i,
# if the L side of the cycle is used.

var p_cycle_usage_span_rest_r{i in SPANS, p in CYCLES} >= 0 <= Large integer;
# Number of unit-capacity copies of cycle p required for restoration of span i,
# if the R side of the cycle is used.

##########################################################################################
# OBJECTIVE
##########################################################################################

minimize total_ModuleCost: sum{m in MODULE_TYPES, j in SPANS} ModuleCost[m] * SpanCost[j] * number_modules[j,m];
# Minimize total module cost.

##########################################################################################
# CONSTRAINTS
##########################################################################################

subject to demands_met{r in DEMANDS}:
sum{q in WORK_ROUTES[r]} workflow[r,q] >= DemandUnits[r];
# All demands must be routed.

subject to working_capacity_assignment{j in SPANS}:
sum{r in DEMANDS, q in WORK_ROUTES[r]} ZetaWorkRoute[j,r,q] * workflow[r,q] <= work[j];
# Enough working capacity for all demands to be routed.

subject to full_restoration{i in SPANS}:
sum{p in CYCLES} (Xpi_L[p,i] * p_cycle_usage_span_rest_l[i,p] + Xpi_R[p,i] * p_cycle_usage_span_rest_r[i,p]) >= work[i];
# Sufficient p-cycles to restore all working capacity.

subject to spare_capacity_placement{j in SPANS}:
sum{p in CYCLES} pCrossesj[p,j] * p_cycle_usage[p] <= spare[j];
# Enough spare capacity to form all p-cycles.

subject to modular_provisioning{j in SPANS}:
sum{m in MODULE_TYPES} ModuleSize[m] * number_modules[j,m] >= work[j] + spare[j];
# Capacity to provision working and spare can be assigned only in modular units.

subject to copies_sufficient_l{i in SPANS, p in CYCLES}:
p_cycle_usage_span_rest[i,p] >= p_cycle_usage_span_rest_l[i,p];
# Must have more copies than number of paths on L side, for each failed span.

subject to copies_sufficient_r{i in SPANS, p in CYCLES}:
p_cycle_usage_span_rest[i,p] >= p_cycle_usage_span_rest_r[i,p];
# Must have more copies than number of paths on R side, for each failed span.
```

```
subject to copies_ok_for_all_failures{i in SPANS, p in CYCLES}:
p_cycle_usage[p] >= p_cycle_usage_span_rest[i,p];
# Enough copies of p-cycle p to restore every single span failure.  This will ensure
# that the number of copies of cycle p is greater than the max number required by
# any one failure.

subject to copies_zero_unacceptable_l{i in SPANS, p in CYCLES}:
p_cycle_usage_span_rest_l[i,p] <= Large * Xpi_L[p,i];
# Number of copies is zero if L side path unacceptable.

subject to copies_zero_unacceptable_r{i in SPANS, p in CYCLES}:
p_cycle_usage_span_rest_r[i,p] <= Large * Xpi_R[p,i];
# Number of copies is zero if R side path unacceptable.
```

3. *p*-Cycle design with multiple Quality of Protection Service Classes

```
#****************************
# SETS
#****************************

set SPANS;
# Set of all spans.

set PCYCLES;
# Set of all p-cycles.


#****************************
# PARAMETERS
#****************************

param Lambda default 1;
# this is the demand growth multiplier.

param QoPMixGold := 0.5;
param QoPMixPlatinum := 0.2;
param QoPMixEconomy := 0.3;

param Cost{j in SPANS};


param Work{j in SPANS};


param Xpi{p in PCYCLES, i in SPANS} default 0;


param pCrossesj{p in PCYCLES, j in SPANS} := sum{i in SPANS: i = j and Xpi[p,j] = 1} 1;


param Work_gold{i in SPANS} := Work[i]*QoPMixGold;
param Work_platinum{i in SPANS} := Work[i]*QoPMixPlatinum;
param Work_economy{i in SPANS} := Work[i]*QoPMixEconomy;


#****************************
# VARIABLES
#****************************

var p_cycle_usage{p in PCYCLES} >=0 integer, <=10000;
# Number of copies of p-cycle p eventually provisoned.

var spare{j in SPANS} >=0 integer, <=10000;
# Number of spare links placed on span j.

var total_spare >=0, <=100000;
# Total cost of spare capacity.
```

```
#****************************
# OBJECTIVE FUNCTION
#****************************

minimize TotalCost: total_spare;
# Minimize the total cost of capacity.
# Total costs of working and spare are calculated individually below as variables.
# We do it this way so that we can simply look at the values of those two variables
# to determine the separate costs of working and spare (instead of needing to set up
# a spreadsheet with individual span capacities to calculate them).


#****************************
# CONSTRAINTS
#****************************

subject to full_restoration{i in SPANS}:
sum{p in PCYCLES}p_cycle_usage[p]* Xpi[p,i]>=Work_gold[i]*Lambda + 2*Work_platinum[i]*Lambda;

subject to plat_rest{i in SPANS}:
sum{p in PCYCLES} Xpi[p,i] * (1-pCrossesj[p,i]) *p_cycle_usage[p] >= 2*Work_platinum[i]*Lambda;

subject to spare_capacity_placement{j in SPANS}:
spare[j] = sum{p in PCYCLES} (pCrossesj[p,j] * p_cycle_usage[p] - Work_economy[j]*Lambda);
# Enough spare capacity is placed on each span to accomodate all the p-cycles
# placed on it.

subject to calulate_spare_cost:
total_spare = sum{j in SPANS} spare[j];
# The total cost of spare capacity is the sum of the costs of spare on each span.
```

4. FIPP *p*-Cycles: (JCP-DRS, SCP, JCP-DRS with Autoformat)

```
##################################################################
# Model
##################################################################
#****************************
# SETS
#****************************
set SPANS;
# Set of all spans.

set DRS;
# set of all DRSs.

set CYCLES;
# Set of all eligible cycles.

set DEMANDS;
# Set of all O-D demand pairs.

set WORK_ROUTES{r in DEMANDS};
# Set of all working routes for each demand pair r.

set ALLWORKINGROUTES;
# set of all working routes for all demands. (big set)

#****************************
# PARAMETERS
#****************************

param Cost{j in SPANS} default 1;
# Cost of installing one module on span j. This is
# a constant cost irrespective of module size.
# This can be used to code in the span length,
# specific geographic/installation/leasing costs etc.

param DemandUnits{r in DEMANDS} default 0;
# units of demand on relation R;

param X_p_i{x in CYCLES, i in SPANS} default 0;
# Number of protection paths a single copy of p-cycle x provides
# span i (2 if straddling span, 1 if on-cycle span, 0 otherwise).
# this has to be provided in the dat file. (can be 0 1 or 2)

param X_p_q{p in CYCLES, q in ALLWORKINGROUTES} default 0;
# Number of protection paths a single copy of p-cycle x provides to
# working route q. (can be 0 1 or 2)

param X_p_c{p in CYCLES, c in DRS} default 0;
# Equal to 1 if DRS c can be protected by eligible cycle p (can be 0 or 1)

param X_c_q{c in DRS, q in ALLWORKINGROUTES} default 0;
# Equal to 1 if DRS c includes route q. (can be 0 or 1)

param PCrossesj{p in CYCLES, j in SPANS} :=
if(X_p_i[p,j]=1) then 1 else 0;
# this simply defines whether a particular span j is on
# cycle p. (can be 0 or 1)

param ZetaWorkRoute{j in SPANS, r in DEMANDS, q in WORK_ROUTES[r]} default 0;
# Equal to 1 1f qth working route for demand between node pair r
# uses span j and 0 otherwise. (can be 0 or 1)

param Omega{q in ALLWORKINGROUTES} default 0;
# Equal to 1 if route q is part of atleast 1 DRS.
# this prevents allocating working capacity to a route
# that can not be protected. (can be 0 or 1)


#****************************
# VARIABLES
#****************************

var spare{j in SPANS} >=0 integer, <=10000;
# Spare capacity placed on span j.

var work{j in SPANS} >=0 integer, <=10000;
# Working capacity placed on span j.

var p_cycle_usage_DRS_route{p in CYCLES, c in DRS, q in ALLWORKINGROUTES} >=0 integer, <=10000;
# Number of copies of p-cycle p required to protect route q of demand r as part of DRS c.

var p_cycle_usage_DRS{p in CYCLES, c in DRS} >=0 integer, <=10000;
# Number of copies of p-cycle p required to protect DRS c.

var p_cycle_usage{p in CYCLES} >=0 integer, <=10000;
# number of copies of p-cycle p actually built.

var workflow{r in DEMANDS, q in WORK_ROUTES[r]} >=0, <=10000;
# Working capacity required by qth working route for demand between node pair r.
# note integrality has been relaxed - as it is implicitly imposed.

#****************************
# OBJECTIVE FUNCTION
#****************************

minimize TotalCost: sum{j in SPANS} Cost[j]*(spare[j]+work[j]);
# minimize the total cost of the modules placed on all spans in the network.

#****************************
# CONSTRAINTS
#****************************
```

```
# Route all working demand across the network.
# do not use working routes that dont belong to atleast one DRS.
subject to demands_met{r in DEMANDS}:
sum{q in WORK_ROUTES[r]} workflow[r,q] * Omega[q] >= DemandUnits[r];


# Place enough working capacity to support all work routing.
# There must be enough working capacity on span j to accomodate all working flows
# simultaneously routed over it by all demand pairs
subject to working_capacity_assignment{j in SPANS}:
work[j] >= sum{r in DEMANDS, q in WORK_ROUTES[r]: ZetaWorkRoute[j,r,q]=1} workflow[r,q];


# Provide sufficent protection relationships for all in-use work routes.
subject to full_route_restoration{r in DEMANDS,q in WORK_ROUTES[r]}:
sum{p in CYCLES, c in DRS: X_c_q[c,q] = 1 and X_p_c[p,c] = 1}
X_p_q[p,q] * p_cycle_usage_DRS_route[p,c,q] >= workflow[r,q];


# Calculates the total number of copies of p-cycle p per DRS c.
subject to p_cycle_usage_per_route{q in ALLWORKINGROUTES, c in DRS,
p in CYCLES: X_c_q[c,q] = 1 and X_p_c[p,c] = 1}:
p_cycle_usage_DRS[p,c] >= p_cycle_usage_DRS_route[p,c,q];


# Calculates the total number of copies of p-cycle p actually built.
subject to p_cycle_usage_per_route_per_DRS{p in CYCLES}:
p_cycle_usage[p] = sum{c in DRS} p_cycle_usage_DRS[p,c],


# Sufficient spare capacity must be placed on each span to
# simultaneously accommodate all p-cycles used.
subject to spare_capacity_placement{j in SPANS}:
spare[j] >= sum{p in CYCLES} PCrossesj[p,j] * p_cycle_usage[p];
```

5. FIPP SCP- Model (from [KoGrJLT])

```
set SPANS;
# (set of all spans)

set NODES,
# (set of all nodes)
set DEMANDS;
# (set of all demand pairs or node pairs)

set PCYCLES;
# set of all eligible cycles.

set WORK_ROUTES{r in DEMANDS};

param Large default 100000;
# A large positive constant (100000).

param Small default 0.01;
# A small positive constant

param Cost{j in SPANS} default 1;
# (cost of each wavelength of on span j)

param DemandUnits{r in DEMANDS} default 0;
# (number of demand units between node pair r)

param DeltaMN{m in DEMANDS, n in DEMANDS} default 0;
# encodes demand rivalry between M and N demands
```

```
param Xpr{p in PCYCLES, r in DEMANDS} default 0;
# is a 2 if cycle p provides 2 protection options to demand r

param Xpi{p in PCYCLES, i in SPANS} default 0;
# is a 2 if cycle p provides 2 protection options to span i
# included to calculate pCrossesj

param pCrossesj{p in PCYCLES, j in SPANS} := sum{i in SPANS: i = j and Xpi[p,j] = 1} 1;
# Equal to 1 if p-cycle p passes over span j, 0 otherwise.
# i.e. if Xpi[p,j] = 1, then p-cycle p crosses span j.


param Work{i in SPANS};
# total work capacity on span i.


# below parameters for compatibility
param DemandRoutingBySpans{r in DEMANDS,i in SPANS} default 0;
param DemandRoutingByNodes{r in DEMANDS,i in NODES} default 0;
param SpanEndNodes{i in SPANS, n in NODES} default 0;
param SpansConnectedToNodes{n in NODES, i in SPANS} default 0;
param ZetaWorkRoute{i in SPANS, r in DEMANDS, k in WORK_ROUTES[r]} default 0;
param SpansCrossedByCycle{p in PCYCLES, i in SPANS} default 0;
param NodesCrossedByCycle{p in PCYCLES, i in NODES} default 0;

var spare{j in SPANS} >=0, <=10000 integer;
# (number of spare wavelengths placed on span j)

var np{p in PCYCLES} >=0, <=10000 integer;
# number of copies of cycle p

var npr{p in PCYCLES, r in DEMANDS} >=0, <=10000;
# number of copies of cycle p for demand r.

var gammapr{p in PCYCLES, r in DEMANDS} >=0, integer <=1;


minimize TotSpare: sum{j in SPANS} spare[j];
# Minimize total spare cost.

subject to protect_all_demands{r in DEMANDS}:
sum{p in PCYCLES}Xpr[p,r]*npr[p,r] >= DemandUnits[r];

subject to max_pcycles_provisioned{p in PCYCLES, r in DEMANDS}:
np[p]>=npr[p,r];

subject to sufficient_spare{j in SPANS}:
spare[j] >= sum{p in PCYCLES} np[p] * pCrossesj[p,j];

subject to compat1{p in PCYCLES, r in DEMANDS}:
gammapr[p,r] >= Small * npr[p,r];

subject to compat2{p in PCYCLES, r in DEMANDS}:
gammapr[p,r] <= Large * npr[p,r];

subject to compat3{p in PCYCLES, m in DEMANDS, n in DEMANDS:m<>n}:
DeltaMN[m,n] + gammapr[p,m] + gammapr[p,n] <= 2;
```

6. FIPP-DRS Model with Auto-Formatting

```
set NETWORK := {"FIPP-cycle-JCP-SeededDRSPropagation_feb32006_Siemens_"};
set CASES := {"30-2-11-1"};
option cplex_options 'mipgap 0.01 presolve 0 display 1 threads 4';


#******************************
# SETS
```

```
#****************************
set SPANS;
# Set of all spans.

set DRS;
# set of all DRSs.

set CYCLES;
# Set of all eligible cycles.

set DEMANDS;
# Set of all O-D demand pairs.

set WORK_ROUTES{r in DEMANDS};
# Set of all working routes for each demand pair r.

set ALLWORKINGROUTES;
# set of all working routes for all demands. (big set)

#****************************
# PARAMETERS
#****************************

param Cost{j in SPANS} default 1;
# Cost of installing one module on span j. This is
# a constant cost irrespective of module size.
# This can be used to code in the span length,
# specific geographic/installation/leasing costs etc.

param DemandUnits{r in DEMANDS} default 0;
# units of demand on relation R;

param X_p_i{x in CYCLES, i in SPANS} default 0;
# Number of protection paths a single copy of p-cycle x provides
# span i (2 if straddling span, 1 if on-cycle span, 0 otherwise).
# this has to be provided in the dat file. (can be 0 1 or 2)

param X_p_q{p in CYCLES, q in ALLWORKINGROUTES} default 0;
# Number of protection paths a single copy of p-cycle x provides to
# working route q. (can be 0 1 or 2)

param X_p_c{p in CYCLES, c in DRS} default 0;
# Equal to 1 if DRS c can be protected by eligible cycle p (can be 0 or 1)

param X_c_q{c in DRS, q in ALLWORKINGROUTES} default 0;
# Equal to 1 if DRS c includes route q. (can be 0 or 1)

param PCrossesj{p in CYCLES, j in SPANS} :=
if(X_p_i[p,j]=1) then 1 else 0;
# this simply defines whether a particular span j is on
# cycle p. (can be 0 or 1)

param ZetaWorkRoute{j in SPANS, r in DEMANDS, q in WORK_ROUTES[r]} default 0;
# Equal to 1 if qth working route for demand between node pair r
# uses span j and 0 otherwise. (can be 0 or 1)

param Omega{q in ALLWORKINGROUTES} default 0;
# Equal to 1 if route q is part of atleast 1 DRS.
# this prevents allocating working capacity to a route
# that can not be protected. (can be 0 or 1)


#****************************
# VARIABLES
#****************************

var spare{j in SPANS} >=0 integer, <=10000;
# Spare capacity placed on span j.
```

```
var work{j in SPANS} >=0 integer, <=10000;
# Working capacity placed on span j.

var p_cycle_usage_DRS_route{p in CYCLES, c in DRS, q in ALLWORKINGROUTES} >=0 integer, <=10000;
# Number of copies of p-cycle p required to protect route q of demand r as part of DRS c.

var p_cycle_usage_DRS{p in CYCLES, c in DRS} >=0 integer, <=10000;
# Number of copies of p-cycle p required to protect DRS c.

var p_cycle_usage{p in CYCLES} >=0 integer, <=10000;
# number of copies of p-cycle p actually built.

var workflow{r in DEMANDS, q in WORK_ROUTES[r]} >=0 integer, <=10000;
# Working capacity required by qth working route for demand between node pair r.
# note integrality has been relaxed - as it is implicitly imposed.

var TotalWorkingCapacity;
var TotalSpareCapacity;
var TotalDesignCost;
var TotalWorkingCost;
var TotalSpareCost;
# both variables above defined for formatting.

#***************************
# OBJECTIVE FUNCTION
#***************************

minimize TotalCost: sum{j in SPANS} Cost[j]*(spare[j]+work[j]);
# minimize the total cost of the modules placed on all spans in the network.

#***************************
# CONSTRAINTS
#***************************


# Route all working demand across the network.
# do not use working routes that dont belong to atleast one DRS.
subject to demands_met{r in DEMANDS}:
sum{q in WORK_ROUTES[r]} workflow[r,q] * Omega[q] >= DemandUnits[r];


# Place enough working capacity to support all work routing.
# There must be enough working capacity on span j to accomodate all working flows
# simultaneously routed over it by all demand pairs
subject to working_capacity_assignment{j in SPANS}:
work[j] >= sum{r in DEMANDS, q in WORK_ROUTES[r]: ZetaWorkRoute[j,r,q]=1} workflow[r,q];


# Provide sufficent protection relationships for all in-use work routes.
subject to full_route_restoration{r in DEMANDS,q in WORK_ROUTES[r]}:
sum{p in CYCLES, c in DRS: X_c_q[c,q] = 1 and X_p_c[p,c] = 1}
X_p_q[p,q] * p_cycle_usage_DRS_route[p,c,q] >= workflow[r,q];


# Calculates the total number of copies of p-cycle p per DRS c.
subject to p_cycle_usage_per_route{q in ALLWORKINGROUTES, c in DRS,
p in CYCLES: X_c_q[c,q] = 1 and X_p_c[p,c] = 1}:
p_cycle_usage_DRS[p,c] >= p_cycle_usage_DRS_route[p,c,q];


# Calculates the total number of copies of p-cycle p actually built.
subject to p_cycle_usage_per_route_per_DRS{p in CYCLES}:
p_cycle_usage[p] = sum{c in DRS} p_cycle_usage_DRS[p,c];


# Sufficient spare capacity must be placed on each span to
# simultaneously accommodate all p-cycles used.
```

```
subject to spare_capacity_placement{j in SPANS}:
spare[j] >= sum{p in CYCLES} PCrossesj[p,j] * p_cycle_usage[p];



 for {c in CASES}{
   for {n in NETWORK}{

   reset data;

   option cplex_options 'mipgap 0.01 presolve 0 display 1 threads 4 timing 1';
   option log_file (n & c & ".log.txt");
   option show_stats 1;

   data (n & c & ".dat");
   solve;
   let TotalWorkingCapacity :=0;
   let TotalSpareCapacity :=0;
   for {i in SPANS} {
      if work[i] >0 then
         {
         printf "work:%s:%d\n",i,work[i] >>  ("SolutionFile." & n & c & ".txt");
         let TotalWorkingCapacity := TotalWorkingCapacity + work[i];
         }


   }

   for {i in SPANS} {
      if spare[i] >0 then
         {
         printf "spare:%s:%d\n",i,spare[i] >>  ("SolutionFile." & n & c & ".txt");
         let TotalSpareCapacity := TotalSpareCapacity + spare[i];
         }

   }

   let TotalDesignCost := sum{j in SPANS} Cost[j]*(spare[j]+work[j]);
   let TotalWorkingCost := sum{j in SPANS} Cost[j]*(work[j]);
   let TotalSpareCost := sum{j in SPANS} Cost[j]*(spare[j]);

   printf "cost_redundancy:%f\n",TotalSpareCost/TotalWorkingCost >> ("SolutionFile." & n & c & ".txt");
   printf "working cost:%f\n",TotalWorkingCost >> ("SolutionFile." & n & c & ".txt");
   printf "spare cost:%f\n",TotalSpareCost >> ("SolutionFile." & n & c & ".txt");
   printf "totalcost:%f\n",TotalDesignCost >> ("SolutionFile." & n & c & ".txt");

   printf "hop_redundancy:%f\n",TotalSpareCapacity/TotalWorkingCapacity >> ("SolutionFile." & n & c & ".txt");
   printf "total_working:%f\n",TotalWorkingCapacity >> ("SolutionFile." & n & c & ".txt");
   printf "total_spare:%f\n",TotalSpareCapacity >> ("SolutionFile." & n & c & ".txt");
   printf "total_capacity_used:%f\n",TotalWorkingCapacity+TotalSpareCapacity >> ("SolutionFile." & n & c & ".txt");

   for {p in CYCLES} {
      if p_cycle_usage[p]>0 then
         printf "cycleusage:%s:%f\n",p,p_cycle_usage[p] >>  ("SolutionFile." & n & c & ".txt");

   }

   for {p in CYCLES} {
      for {d in DRS} {
         if p_cycle_usage_DRS[p,d] >0.1 then
         printf "cycleusagedrs:%s:%s:%f\n",p,d,p_cycle_usage_DRS[p,d] >>  ("SolutionFile." & n & c & ".txt");
      }

   }

   for {p in CYCLES} {
      for {d in DRS} {
        for {q in ALLWORKINGROUTES}{
        if p_cycle_usage_DRS_route[p,d,q] >0.1 then
        printf "cycleusagedrsroute:%s:%s:%s:%f:%s\n",p,d,q,p_cycle_usage_DRS_route[p,d,q],X_p_q[p,q] >> ("SolutionFile." & n
& c & ".txt");
        }
      }

  }

  for {r in DEMANDS} {
     for {q in WORK_ROUTES[r]}{
        if workflow[r,q]>0.1 then
        printf "workflow:%s:%s:%f\n",r,q,workflow[r,q] >> ("SolutionFile." & n & c & ".txt");
     }


  }

  for {r in DEMANDS} {
     for {q in WORK_ROUTES[r]}{
        for {p in CYCLES}{
           for {d in DRS}{
              if workflow[r,q]>0.1 and p_cycle_usage_DRS_route[p,d,q]>0.1 then
              printf "demandroutecycle:%s:%s:%s:%f:%s:%f\n",r,DemandUnits[r],q,workflow[r,q],p,p_cycle_usage_DRS_route[p,d,q]
>> ("SolutionFile." & n & c & ".txt");
              }
           }
        }
     }

  }


}
```

Appendix 2. Test Networks

1. COST 239 Network

Topology

NODE	X	Y	SIZE
N0	140	281	6
N1	315	350	4
N2	304	298	5
N3	356	235	5
N4	447	308	4
N5	417	161	5
N6	242	159	5
N7	250	214	5
N8	185	208	5
N9	128	143	4
N10	344	50	4

SPAN	O	D	LENGTH	UNITCOST
S1	N0	N1	820	820
S2	N0	N2	600	600
S3	N0	N5	1090	1090
S4	N0	N7	400	400
S5	N0	N8	300	300
S6	N0	N9	450	450

S7	N1	N2	320	320
S8	N1	N4	820	820
S9	N1	N8	930	930
S10	N2	N3	565	565
S11	N2	N4	730	730
S12	N2	N7	350	350
S13	N3	N10	740	740
S14	N3	N4	320	320
S15	N3	N5	340	340
S16	N3	N7	730	730
S17	N4	N5	660	660
S18	N5	N10	390	390
S19	N5	N6	660	660
S20	N6	N10	760	760
S21	N6	N7	390	390
S22	N6	N8	210	210
S23	N6	N9	550	550
S24	N7	N8	220	220
S25	N8	N9	390	390
S26	N9	N10	1310	1310

Demands (55 Demands)

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
0	0	1	2	R1	inf	inf	0
1	0	2	2	R1	inf	inf	0
2	0	3	2	R1	inf	inf	0
3	0	4	2	R1	inf	inf	0
4	0	5	2	R1	inf	inf	0
5	0	6	2	R1	inf	inf	0
6	0	7	2	R1	inf	inf	0
7	0	8	2	R1	inf	inf	0
8	0	9	2	R1	inf	inf	0
9	0	10	2	R1	inf	inf	0
10	1	2	2	R1	inf	inf	0
11	1	3	2	R1	inf	inf	0
12	1	4	2	R1	inf	inf	0
13	1	5	2	R1	inf	inf	0
14	1	6	2	R1	inf	inf	0
15	1	7	2	R1	inf	inf	0
16	1	8	2	R1	inf	inf	0
17	1	9	2	R1	inf	inf	0
18	1	10	2	R1	inf	inf	0
19	2	3	2	R1	inf	inf	0
20	2	4	2	R1	inf	inf	0
21	2	5	2	R1	inf	inf	0
22	2	6	2	R1	inf	inf	0
23	2	7	2	R1	inf	inf	0
24	2	8	2	R1	inf	inf	0
25	2	9	2	R1	inf	inf	0
26	2	10	2	R1	inf	inf	0
27	3	4	2	R1	inf	inf	0
28	3	5	2	R1	inf	inf	0
29	3	6	2	R1	inf	inf	0
30	3	7	2	R1	inf	inf	0
31	3	8	2	R1	inf	inf	0
32	3	9	2	R1	inf	inf	0
33	3	10	2	R1	inf	inf	0
34	4	5	2	R1	inf	inf	0
35	4	6	2	R1	inf	inf	0
36	4	7	2	R1	inf	inf	0
37	4	8	2	R1	inf	inf	0
38	4	9	2	R1	inf	inf	0
39	4	10	2	R1	inf	inf	0
40	5	6	2	R1	inf	inf	0
41	5	7	2	R1	inf	inf	0
42	5	8	2	R1	inf	inf	0
43	5	9	2	R1	inf	inf	0

44	5	10	2	R1	inf	inf	0
45	6	7	2	R1	inf	inf	0
46	6	8	2	R1	inf	inf	0
47	6	9	2	R1	inf	inf	0
48	6	10	2	R1	inf	inf	0
49	7	8	2	R1	inf	inf	0
50	7	9	2	R1	inf	inf	0
51	7	10	2	R1	inf	inf	0
52	8	9	2	R1	inf	inf	0
53	8	10	2	R1	inf	inf	0
54	9	10	2	R1	inf	inf	0

Demands (27 Demands)

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
0	0	2	2	R1	inf	inf	0
1	0	4	2	R1	inf	inf	0
2	0	6	2	R1	inf	inf	0
3	0	8	2	R1	inf	inf	0
4	0	10	2	R1	inf	inf	0
5	1	3	2	R1	inf	inf	0
6	1	5	2	R1	inf	inf	0
7	1	7	2	R1	inf	inf	0
8	1	9	2	R1	inf	inf	0
9	2	3	2	R1	inf	inf	0
10	2	5	2	R1	inf	inf	0
11	2	7	2	R1	inf	inf	0
12	2	9	2	R1	inf	inf	0
13	3	4	2	R1	inf	inf	0
14	3	6	2	R1	inf	inf	0
15	3	8	2	R1	inf	inf	0
16	3	10	2	R1	inf	inf	0
17	4	6	2	R1	inf	inf	0
18	4	8	2	R1	inf	inf	0
19	4	10	2	R1	inf	inf	0
20	5	7	2	R1	inf	inf	0
21	5	9	2	R1	inf	inf	0
22	6	7	2	R1	inf	inf	0
23	6	9	2	R1	inf	inf	0
24	7	8	2	R1	inf	inf	0
25	7	10	2	R1	inf	inf	0
26	8	10	2	R1	inf	inf	0

2. 13n23s Network

Topology

NODE	X	Y	SIZE
N02	268	55	4
N04	525	97	3
N05	290	233	6
N06	612	218	5
N07	508	349	4
N08	572	485	4
N09	402	456	4
N10	259	325	6
N11	291	598	3
N12	292	483	4
N13	186	458	3
N14	49	421	4
N15	75	274	3

SPAN	O	D	LENGTH	UNITCOST
S05	N02	N04	260.409	260.409
S06	N02	N05	179.354	179.354
S10	N04	N06	149.03	149.03

S11	N05	N07	246.941	246.941
S12	N05	N10	97.082	97.082
S13	N05	N15	218.874	218.874
S14	N06	N07	167.263	167.263
S16	N07	N09	150.615	150.615
S17	N08	N06	269.98	269.98
S18	N08	N11	302.87	302.87
S19	N09	N08	172.456	172.456
S21	N10	N06	368.86	368.86
S23	N11	N12	115.004	115.004
S24	N11	N14	299.822	299.822
S25	N12	N09	113.265	113.265
S26	N12	N10	161.409	161.409
S27	N13	N10	151.717	151.717
S28	N13	N12	108.908	108.908
S29	N14	N13	141.908	141.908
S30	N15	N14	149.282	149.282
S31	N15	N02	291.908	291.908
S32	N05	N04	271.516	271.516
S33	N02	N06	380.664	380.664

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
N02-N04	N02	N04	9	R1	inf	inf	0
N02-N05	N02	N05	4	R1	inf	inf	0
N02-N06	N02	N06	1	R1	inf	inf	0
N02-N07	N02	N07	7	R1	inf	inf	0
N02-N08	N02	N08	3	R1	inf	inf	0
N02-N09	N02	N09	4	R1	inf	inf	0
N02-N10	N02	N10	6	R1	inf	inf	0
N02-N11	N02	N11	2	R1	inf	inf	0
N02-N12	N02	N12	2	R1	inf	inf	0
N02-N13	N02	N13	2	R1	inf	inf	0
N02-N14	N02	N14	5	R1	inf	inf	0
N02-N15	N02	N15	2	R1	inf	inf	0
N04-N05	N04	N05	1	R1	inf	inf	0
N04-N06	N04	N06	2	R1	inf	inf	0
N04-N07	N04	N07	1	R1	inf	inf	0
N04-N08	N04	N08	6	R1	inf	inf	0
N04-N09	N04	N09	3	R1	inf	inf	0
N04-N10	N04	N10	1	R1	inf	inf	0
N04-N11	N04	N11	9	R1	inf	inf	0
N04-N12	N04	N12	8	R1	inf	inf	0
N04-N13	N04	N13	4	R1	inf	inf	0
N04-N14	N04	N14	5	R1	inf	inf	0
N04-N15	N04	N15	4	R1	inf	inf	0
N05-N06	N05	N06	3	R1	inf	inf	0
N05-N07	N05	N07	8	R1	inf	inf	0
N05-N08	N05	N08	8	R1	inf	inf	0
N05-N09	N05	N09	7	R1	inf	inf	0
N05-N10	N05	N10	4	R1	inf	inf	0
N05-N11	N05	N11	6	R1	inf	inf	0
N05-N12	N05	N12	8	R1	inf	inf	0
N05-N13	N05	N13	1	R1	inf	inf	0
N05-N14	N05	N14	8	R1	inf	inf	0
N05-N15	N05	N15	9	R1	inf	inf	0
N06-N07	N06	N07	5	R1	inf	inf	0
N06-N08	N06	N08	9	R1	inf	inf	0
N06-N09	N06	N09	8	R1	inf	inf	0
N06-N10	N06	N10	9	R1	inf	inf	0
N06-N11	N06	N11	6	R1	inf	inf	0
N06-N12	N06	N12	7	R1	inf	inf	0
N06-N13	N06	N13	4	R1	inf	inf	0
N06-N14	N06	N14	2	R1	inf	inf	0
N06-N15	N06	N15	3	R1	inf	inf	0
N07-N08	N07	N08	1	R1	inf	inf	0
N07-N09	N07	N09	5	R1	inf	inf	0

N07-N10	N07	N10	6	R1	inf	inf	0
N07-N11	N07	N11	1	R1	inf	inf	0
N07-N12	N07	N12	2	R1	inf	inf	0
N07-N13	N07	N13	1	R1	inf	inf	0
N07-N14	N07	N14	8	R1	inf	inf	0
N07-N15	N07	N15	4	R1	inf	inf	0
N08-N09	N08	N09	3	R1	inf	inf	0
N08-N10	N08	N10	6	R1	inf	inf	0
N08-N11	N08	N11	6	R1	inf	inf	0
N08-N12	N08	N12	6	R1	inf	inf	0
N08-N13	N08	N13	9	R1	inf	inf	0
N08-N14	N08	N14	5	R1	inf	inf	0
N08-N15	N08	N15	9	R1	inf	inf	0
N09-N10	N09	N10	1	R1	inf	inf	0
N09-N11	N09	N11	8	R1	inf	inf	0
N09-N12	N09	N12	8	R1	inf	inf	0
N09-N13	N09	N13	9	R1	inf	inf	0
N09-N14	N09	N14	4	R1	inf	inf	0
N09-N15	N09	N15	7	R1	inf	inf	0
N10-N11	N10	N11	4	R1	inf	inf	0
N10-N12	N10	N12	2	R1	inf	inf	0
N10-N13	N10	N13	5	R1	inf	inf	0
N10-N14	N10	N14	7	R1	inf	inf	0
N10-N15	N10	N15	7	R1	inf	inf	0
N11-N12	N11	N12	9	R1	inf	inf	0
N11-N13	N11	N13	2	R1	inf	inf	0
N11-N14	N11	N14	5	R1	inf	inf	0
N11-N15	N11	N15	7	R1	inf	inf	0
N12-N13	N12	N13	5	R1	inf	inf	0
N12-N14	N12	N14	5	R1	inf	inf	0
N12-N15	N12	N15	6	R1	inf	inf	0
N13-N14	N13	N14	1	R1	inf	inf	0
N13-N15	N13	N15	7	R1	inf	inf	0
N14-N15	N14	N15	3	R1	inf	inf	0

3. 12n19s Network

Topology

NODE	X	Y	SIZE
N02	268	55	4
N04	525	97	3
N05	290	233	6
N06	612	218	5
N07	508	349	4
N08	572	485	4
N09	402	456	4
N11	291	598	3
N12	292	483	4
N13	186	458	3
N14	49	421	4
N15	75	274	3

SPAN	O	D	LENGTH	UNITCOST
S05	N02	N04	260.409	260.409
S06	N02	N05	179.354	179.354
S10	N04	N06	149.03	149.03
S11	N05	N07	246.941	246.941
S14	N06	N07	167.263	167.263
S16	N07	N09	150.615	150.615
S17	N08	N06	269.98	269.98
S18	N08	N11	302.87	302.87
S19	N09	N08	172.456	172.456
S23	N11	N12	115.004	115.004
S24	N11	N14	299.822	299.822
S25	N12	N09	113.265	113.265
S28	N13	N12	108.908	108.908

S29	N14	N13	141.908	141.908
S30	N15	N14	149.282	149.282
S31	N15	N02	291.908	291.908
S32	N05	N04	271.516	271.516
S33	N15	N05	218.874	218.874
S34	N13	N05	247.873	247.873

Demands

```
DEMAND O  D   NBUNITS RESTCLASS HOPLIM DISTLIM MAXOUTAGE
N02-N04 N02 N04 5     R1    inf  inf  0
N02-N05 N02 N05 6     R1    inf  inf  0
N02-N06 N02 N06 10    R1    inf  inf  0
N02-N07 N02 N07 7     R1    inf  inf  0
N02-N08 N02 N08 9     R1    inf  inf  0
N02-N09 N02 N09 9     R1    inf  inf  0
N02-N11 N02 N11 4     R1    inf  inf  0
N02-N12 N02 N12 8     R1    inf  inf  0
N02-N13 N02 N13 8     R1    inf  inf  0
N02-N14 N02 N14 4     R1    inf  inf  0
N02-N15 N02 N15 3     R1    inf  inf  0
N04-N05 N04 N05 8     R1    inf  inf  0
N04-N06 N04 N06 7     R1    inf  inf  0
N04-N07 N04 N07 2     R1    inf  inf  0
N04-N08 N04 N08 10    R1    inf  inf  0
N04-N09 N04 N09 10    R1    inf  inf  0
N04-N11 N04 N11 9     R1    inf  inf  0
N04-N12 N04 N12 5     R1    inf  inf  0
N04-N13 N04 N13 5     R1    inf  inf  0
N04-N14 N04 N14 5     R1    inf  inf  0
N04-N15 N04 N15 10    R1    inf  inf  0
N05-N06 N05 N06 3     R1    inf  inf  0
N05-N07 N05 N07 6     R1    inf  inf  0
N05-N08 N05 N08 3     R1    inf  inf  0
N05-N09 N05 N09 2     R1    inf  inf  0
N05-N11 N05 N11 9     R1    inf  inf  0
N05-N12 N05 N12 2     R1    inf  inf  0
N05-N13 N05 N13 7     R1    inf  inf  0
N05-N14 N05 N14 8     R1    inf  inf  0
N05-N15 N05 N15 3     R1    inf  inf  0
N06-N07 N06 N07 5     R1    inf  inf  0
N06-N08 N06 N08 8     R1    inf  inf  0
N06-N09 N06 N09 5     R1    inf  inf  0
N06-N11 N06 N11 4     R1    inf  inf  0
N06-N12 N06 N12 3     R1    inf  inf  0
N06-N13 N06 N13 6     R1    inf  inf  0
N06-N14 N06 N14 7     R1    inf  inf  0
N06-N15 N06 N15 4     R1    inf  inf  0
N07-N08 N07 N08 2     R1    inf  inf  0
N07-N09 N07 N09 7     R1    inf  inf  0
N07-N11 N07 N11 6     R1    inf  inf  0
N07-N12 N07 N12 2     R1    inf  inf  0
N07-N13 N07 N13 8     R1    inf  inf  0
N07-N14 N07 N14 9     R1    inf  inf  0
N07-N15 N07 N15 10    R1    inf  inf  0
N08-N09 N08 N09 3     R1    inf  inf  0
N08-N11 N08 N11 9     R1    inf  inf  0
N08-N12 N08 N12 9     R1    inf  inf  0
N08-N13 N08 N13 3     R1    inf  inf  0
N08-N14 N08 N14 8     R1    inf  inf  0
N08-N15 N08 N15 9     R1    inf  inf  0
N09-N11 N09 N11 4     R1    inf  inf  0
N09-N12 N09 N12 7     R1    inf  inf  0
N09-N13 N09 N13 8     R1    inf  inf  0
N09-N14 N09 N14 7     R1    inf  inf  0
N09-N15 N09 N15 10    R1    inf  inf  0
N11-N12 N11 N12 6     R1    inf  inf  0
N11-N13 N11 N13 2     R1    inf  inf  0
```

N11-N14	N11	N14	7	R1	inf	inf	0
N11-N15	N11	N15	6	R1	inf	inf	0
N12-N13	N12	N13	4	R1	inf	inf	0
N12-N14	N12	N14	9	R1	inf	inf	0
N12-N15	N12	N15	7	R1	inf	inf	0
N13-N14	N13	N14	8	R1	inf	inf	0
N13-N15	N13	N15	5	R1	inf	inf	0
N14-N15	N14	N15	8	R1	inf	inf	0

4. NSFNET Network

Topology

NODE	X	Y	SIZE
N01	9	126	3
N02	57	69	3
N03	53	193	3
N04	98	117	3
N05	156	138	3
N06	152	211	4
N07	222	118	2
N08	278	84	3
N09	327	48	4
N10	312	237	2
N11	369	9	3
N12	429	53	3
N13	425	138	3
N14	360	188	3

SPAN	O	D	LENGTH	UNITCOST
S01	N01	N02	74.518	74.518
S02	N01	N03	80.156	80.156
S03	N01	N04	89.454	89.454
S04	N02	N03	124.064	124.064
S05	N02	N08	221.508	221.508
S06	N03	N06	100.623	100.623
S07	N04	N05	61.685	61.685
S08	N04	N11	291.728	291.728
S09	N05	N06	73.11	73.11
S10	N05	N07	68.964	68.964
S11	N06	N10	162.099	162.099
S12	N06	N14	209.268	209.268
S13	N07	N08	65.513	65.513
S14	N08	N09	60.803	60.803
S15	N09	N10	189.594	189.594
S16	N09	N12	102.122	102.122
S17	N09	N13	133.056	133.056
S18	N11	N12	74.404	74.404
S19	N11	N13	140.631	140.631
S20	N12	N14	151.611	151.611
S21	N13	N14	82.006	82.006

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
N01-N02	N01	N02	7	Gold	inf	inf	0
N01-N03	N01	N03	2	Gold	inf	inf	0
N01-N04	N01	N04	10	Gold	inf	inf	0
N01-N05	N01	N05	5	Gold	inf	inf	0
N01-N06	N01	N06	5	Gold	inf	inf	0
N01-N07	N01	N07	3	Gold	inf	inf	0
N01-N08	N01	N08	9	Gold	inf	inf	0
N01-N09	N01	N09	8	Gold	inf	inf	0
N01-N10	N01	N10	8	Gold	inf	inf	0
N01-N11	N01	N11	8	Gold	inf	inf	0
N01-N12	N01	N12	5	Gold	inf	inf	0
N01-N13	N01	N13	7	Gold	inf	inf	0

N01-N14	N01	N14	9	Gold	inf	inf	0
N02-N03	N02	N03	7	Gold	inf	inf	0
N02-N04	N02	N04	3	Gold	inf	inf	0
N02-N05	N02	N05	5	Gold	inf	inf	0
N02-N06	N02	N06	9	Gold	inf	inf	0
N02-N07	N02	N07	1	Gold	inf	inf	0
N02-N08	N02	N08	10	Gold	inf	inf	0
N02-N09	N02	N09	3	Gold	inf	inf	0
N02-N10	N02	N10	7	Gold	inf	inf	0
N02-N11	N02	N11	7	Gold	inf	inf	0
N02-N12	N02	N12	6	Gold	inf	inf	0
N02-N13	N02	N13	2	Gold	inf	inf	0
N02-N14	N02	N14	3	Gold	inf	inf	0
N03-N04	N03	N04	5	Gold	inf	inf	0
N03-N05	N03	N05	6	Gold	inf	inf	0
N03-N06	N03	N06	1	Gold	inf	inf	0
N03-N07	N03	N07	1	Gold	inf	inf	0
N03-N08	N03	N08	1	Gold	inf	inf	0
N03-N09	N03	N09	2	Gold	inf	inf	0
N03-N10	N03	N10	1	Gold	inf	inf	0
N03-N11	N03	N11	8	Gold	inf	inf	0
N03-N12	N03	N12	2	Gold	inf	inf	0
N03-N13	N03	N13	5	Gold	inf	inf	0
N03-N14	N03	N14	4	Gold	inf	inf	0
N04-N05	N04	N05	6	Gold	inf	inf	0
N04-N06	N04	N06	7	Gold	inf	inf	0
N04-N07	N04	N07	5	Gold	inf	inf	0
N04-N08	N04	N08	10	Gold	inf	inf	0
N04-N09	N04	N09	5	Gold	inf	inf	0
N04-N10	N04	N10	5	Gold	inf	inf	0
N04-N11	N04	N11	7	Gold	inf	inf	0
N04-N12	N04	N12	8	Gold	inf	inf	0
N04-N13	N04	N13	10	Gold	inf	inf	0
N04-N14	N04	N14	2	Gold	inf	inf	0
N05-N06	N05	N06	7	Gold	inf	inf	0
N05-N07	N05	N07	1	Gold	inf	inf	0
N05-N08	N05	N08	5	Gold	inf	inf	0
N05-N09	N05	N09	1	Gold	inf	inf	0
N05-N10	N05	N10	2	Gold	inf	inf	0
N05-N11	N05	N11	4	Gold	inf	inf	0
N05-N12	N05	N12	7	Gold	inf	inf	0
N05-N13	N05	N13	3	Gold	inf	inf	0
N05-N14	N05	N14	7	Gold	inf	inf	0
N06-N07	N06	N07	8	Gold	inf	inf	0
N06-N08	N06	N08	8	Gold	inf	inf	0
N06-N09	N06	N09	8	Gold	inf	inf	0
N06-N10	N06	N10	5	Gold	inf	inf	0
N06-N11	N06	N11	2	Gold	inf	inf	0
N06-N12	N06	N12	2	Gold	inf	inf	0
N06-N13	N06	N13	2	Gold	inf	inf	0
N06-N14	N06	N14	3	Gold	inf	inf	0
N07-N08	N07	N08	10	Gold	inf	inf	0
N07-N09	N07	N09	3	Gold	inf	inf	0
N07-N10	N07	N10	10	Gold	inf	inf	0
N07-N11	N07	N11	2	Gold	inf	inf	0
N07-N12	N07	N12	7	Gold	inf	inf	0
N07-N13	N07	N13	10	Gold	inf	inf	0
N07-N14	N07	N14	7	Gold	inf	inf	0
N08-N09	N08	N09	10	Gold	inf	inf	0
N08-N10	N08	N10	4	Gold	inf	inf	0
N08-N11	N08	N11	7	Gold	inf	inf	0
N08-N12	N08	N12	1	Gold	inf	inf	0
N08-N13	N08	N13	6	Gold	inf	inf	0
N08-N14	N08	N14	5	Gold	inf	inf	0
N09-N10	N09	N10	4	Gold	inf	inf	0
N09-N11	N09	N11	8	Gold	inf	inf	0
N09-N12	N09	N12	5	Gold	inf	inf	0
N09-N13	N09	N13	4	Gold	inf	inf	0

N09-N14	N09	N14	9	Gold	inf	inf	0
N10-N11	N10	N11	9	Gold	inf	inf	0
N10-N12	N10	N12	5	Gold	inf	inf	0
N10-N13	N10	N13	5	Gold	inf	inf	0
N10-N14	N10	N14	2	Gold	inf	inf	0
N11-N12	N11	N12	1	Gold	inf	inf	0
N11-N13	N11	N13	3	Gold	inf	inf	0
N11-N14	N11	N14	9	Gold	inf	inf	0
N12-N13	N12	N13	5	Gold	inf	inf	0
N12-N14	N12	N14	1	Gold	inf	inf	0
N13-N14	N13	N14	9	Gold	inf	inf	0

5. 19n35s1 Network

Topology

NODE	X	Y	SIZE
N01	183	456	3
N02	222	322	3
N03	275	163	4
N04	266	297	5
N06	470	253	4
N07	378	241	5
N08	331	314	4
N09	476	318	5
N10	607	311	3
N11	533	410	5
N12	634	482	3
N13	513	516	4
N14	406	389	6
N15	285	437	4
N16	338	473	5
N17	433	538	3
N18	510	637	3
N19	380	549	4
N20	260	543	3

SPAN	O	D	LENGTH	UNITCOST
S01	N01	N02	139.56	139.56
S03	N01	N20	116.181	116.181
S04	N02	N03	167.601	167.601
S05	N02	N04	50.606	50.606
S07	N04	N03	134.302	134.302
S13	N06	N10	148.772	148.772
S15	N07	N06	92.779	92.779
S16	N07	N08	86.822	86.822
S17	N07	N09	124.631	124.631
S18	N08	N09	145.055	145.055
S19	N08	N15	131.32	131.32
S20	N09	N11	108.227	108.227
S21	N09	N14	99.705	99.705
S22	N10	N09	131.187	131.187
S23	N10	N12	173.118	173.118
S24	N11	N12	124.036	124.036
S25	N11	N13	107.87	107.87
S26	N12	N18	198.497	198.497
S27	N13	N17	82.97	82.97
S32	N15	N16	64.07	64.07
S33	N16	N11	204.924	204.924
S34	N16	N14	108.074	108.074
S35	N16	N20	104.805	104.805
S36	N17	N19	54.129	54.129
S37	N18	N13	121.037	121.037
S38	N19	N16	86.833	86.833
S39	N19	N18	156.984	156.984
S40	N20	N19	120.15	120.15
S41	N03	N07	129.201	129.201

S42	N04	N20	246.073	246.073
S43	N15	N01	103.755	103.755
S44	N14	N17	151.427	151.427
N03-N08	N03	N08	161.05	161.05
N04-N08	N04	N08	67.186	67.186
N09-N06	N09	N06	65.276	65.276

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
N01-N02	N01	N02	7	R1	inf	inf	0
N01-N03	N01	N03	8	R1	inf	inf	0
N01-N04	N01	N04	7	R1	inf	inf	0
N01-N06	N01	N06	3	R1	inf	inf	0
N01-N07	N01	N07	8	R1	inf	inf	0
N01-N08	N01	N08	9	R1	inf	inf	0
N01-N09	N01	N09	8	R1	inf	inf	0
N01-N10	N01	N10	2	R1	inf	inf	0
N01-N11	N01	N11	4	R1	inf	inf	0
N01-N12	N01	N12	2	R1	inf	inf	0
N01-N13	N01	N13	5	R1	inf	inf	0
N01-N14	N01	N14	9	R1	inf	inf	0
N01-N15	N01	N15	2	R1	inf	inf	0
N01-N16	N01	N16	10	R1	inf	inf	0
N01-N17	N01	N17	10	R1	inf	inf	0
N01-N18	N01	N18	5	R1	inf	inf	0
N01-N19	N01	N19	2	R1	inf	inf	0
N01-N20	N01	N20	5	R1	inf	inf	0
N02-N03	N02	N03	3	R1	inf	inf	0
N02-N04	N02	N04	4	R1	inf	inf	0
N02-N06	N02	N06	8	R1	inf	inf	0
N02-N07	N02	N07	10	R1	inf	inf	0
N02-N08	N02	N08	3	R1	inf	inf	0
N02-N09	N02	N09	7	R1	inf	inf	0
N02-N10	N02	N10	6	R1	inf	inf	0
N02-N11	N02	N11	4	R1	inf	inf	0
N02-N12	N02	N12	6	R1	inf	inf	0
N02-N13	N02	N13	9	R1	inf	inf	0
N02-N14	N02	N14	9	R1	inf	inf	0
N02-N15	N02	N15	4	R1	inf	inf	0
N02-N16	N02	N16	6	R1	inf	inf	0
N02-N17	N02	N17	2	R1	inf	inf	0
N02-N18	N02	N18	4	R1	inf	inf	0
N02-N19	N02	N19	5	R1	inf	inf	0
N02-N20	N02	N20	9	R1	inf	inf	0
N03-N04	N03	N04	7	R1	inf	inf	0
N03-N06	N03	N06	2	R1	inf	inf	0
N03-N07	N03	N07	6	R1	inf	inf	0
N03-N08	N03	N08	7	R1	inf	inf	0
N03-N09	N03	N09	5	R1	inf	inf	0
N03-N10	N03	N10	8	R1	inf	inf	0
N03-N11	N03	N11	4	R1	inf	inf	0
N03-N12	N03	N12	8	R1	inf	inf	0
N03-N13	N03	N13	4	R1	inf	inf	0
N03-N14	N03	N14	2	R1	inf	inf	0
N03-N15	N03	N15	4	R1	inf	inf	0
N03-N16	N03	N16	6	R1	inf	inf	0
N03-N17	N03	N17	5	R1	inf	inf	0
N03-N18	N03	N18	7	R1	inf	inf	0
N03-N19	N03	N19	4	R1	inf	inf	0
N03-N20	N03	N20	3	R1	inf	inf	0
N04-N06	N04	N06	5	R1	inf	inf	0
N04-N07	N04	N07	9	R1	inf	inf	0
N04-N08	N04	N08	7	R1	inf	inf	0
N04-N09	N04	N09	8	R1	inf	inf	0
N04-N10	N04	N10	6	R1	inf	inf	0
N04-N11	N04	N11	7	R1	inf	inf	0
N04-N12	N04	N12	10	R1	inf	inf	0

```
N04-N13 N04 N13 5    R1    inf  inf  0
N04-N14 N04 N14 3    R1    inf  inf  0
N04-N15 N04 N15 2    R1    inf  inf  0
N04-N16 N04 N16 4    R1    inf  inf  0
N04-N17 N04 N17 4    R1    inf  inf  0
N04-N18 N04 N18 2    R1    inf  inf  0
N04-N19 N04 N19 8    R1    inf  inf  0
N04-N20 N04 N20 9    R1    inf  inf  0
N06-N07 N06 N07 9    R1    inf  inf  0
N06-N08 N06 N08 8    R1    inf  inf  0
N06-N09 N06 N09 3    R1    inf  inf  0
N06-N10 N06 N10 3    R1    inf  inf  0
N06-N11 N06 N11 2    R1    inf  inf  0
N06-N12 N06 N12 7    R1    inf  inf  0
N06-N13 N06 N13 7    R1    inf  inf  0
N06-N14 N06 N14 3    R1    inf  inf  0
N06-N15 N06 N15 4    R1    inf  inf  0
N06-N16 N06 N16 10    R1    inf  inf  0
N06-N17 N06 N17 6    R1    inf  inf  0
N06-N18 N06 N18 9    R1    inf  inf  0
N06-N19 N06 N19 5    R1    inf  inf  0
N06-N20 N06 N20 9    R1    inf  inf  0
N07-N08 N07 N08 4    R1    inf  inf  0
N07-N09 N07 N09 2    R1    inf  inf  0
N07-N10 N07 N10 10    R1    inf  inf  0
N07-N11 N07 N11 8    R1    inf  inf  0
N07-N12 N07 N12 7    R1    inf  inf  0
N07-N13 N07 N13 6    R1    inf  inf  0
N07-N14 N07 N14 6    R1    inf  inf  0
N07-N15 N07 N15 4    R1    inf  inf  0
N07-N16 N07 N16 7    R1    inf  inf  0
N07-N17 N07 N17 5    R1    inf  inf  0
N07-N18 N07 N18 5    R1    inf  inf  0
N07-N19 N07 N19 7    R1    inf  inf  0
N07-N20 N07 N20 7    R1    inf  inf  0
N08-N09 N08 N09 6    R1    inf  inf  0
N08-N10 N08 N10 9    R1    inf  inf  0
N08-N11 N08 N11 6    R1    inf  inf  0
N08-N12 N08 N12 3    R1    inf  inf  0
N08-N13 N08 N13 9    R1    inf  inf  0
N08-N14 N08 N14 3    R1    inf  inf  0
N08-N15 N08 N15 6    R1    inf  inf  0
N08-N16 N08 N16 3    R1    inf  inf  0
N08-N17 N08 N17 2    R1    inf  inf  0
N08-N18 N08 N18 5    R1    inf  inf  0
N08-N19 N08 N19 2    R1    inf  inf  0
N08-N20 N08 N20 5    R1    inf  inf  0
N09-N10 N09 N10 6    R1    inf  inf  0
N09-N11 N09 N11 3    R1    inf  inf  0
N09-N12 N09 N12 6    R1    inf  inf  0
N09-N13 N09 N13 9    R1    inf  inf  0
N09-N14 N09 N14 3    R1    inf  inf  0
N09-N15 N09 N15 6    R1    inf  inf  0
N09-N16 N09 N16 8    R1    inf  inf  0
N09-N17 N09 N17 6    R1    inf  inf  0
N09-N18 N09 N18 3    R1    inf  inf  0
N09-N19 N09 N19 8    R1    inf  inf  0
N09-N20 N09 N20 3    R1    inf  inf  0
N10-N11 N10 N11 6    R1    inf  inf  0
N10-N12 N10 N12 8    R1    inf  inf  0
N10-N13 N10 N13 2    R1    inf  inf  0
N10-N14 N10 N14 3    R1    inf  inf  0
N10-N15 N10 N15 7    R1    inf  inf  0
N10-N16 N10 N16 2    R1    inf  inf  0
N10-N17 N10 N17 8    R1    inf  inf  0
N10-N18 N10 N18 7    R1    inf  inf  0
N10-N19 N10 N19 7    R1    inf  inf  0
N10-N20 N10 N20 7    R1    inf  inf  0
```

```
N11-N12 N11 N12 3    R1    inf  inf   0
N11-N13 N11 N13 8    R1    inf  inf   0
N11-N14 N11 N14 7    R1    inf  inf   0
N11-N15 N11 N15 3    R1    inf  inf   0
N11-N16 N11 N16 7    R1    inf  inf   0
N11-N17 N11 N17 3    R1    inf  inf   0
N11-N18 N11 N18 7    R1    inf  inf   0
N11-N19 N11 N19 8    R1    inf  inf   0
N11-N20 N11 N20 7    R1    inf  inf   0
N12-N13 N12 N13 4    R1    inf  inf   0
N12-N14 N12 N14 3    R1    inf  inf   0
N12-N15 N12 N15 4    R1    inf  inf   0
N12-N16 N12 N16 5    R1    inf  inf   0
N12-N17 N12 N17 7    R1    inf  inf   0
N12-N18 N12 N18 5    R1    inf  inf   0
N12-N19 N12 N19 5    R1    inf  inf   0
N12-N20 N12 N20 8    R1    inf  inf   0
N13-N14 N13 N14 9    R1    inf  inf   0
N13-N15 N13 N15 10    R1    inf  inf   0
N13-N16 N13 N16 4    R1    inf  inf   0
N13-N17 N13 N17 9    R1    inf  inf   0
N13-N18 N13 N18 5    R1    inf  inf   0
N13-N19 N13 N19 8    R1    inf  inf   0
N13-N20 N13 N20 9    R1    inf  inf   0
N14-N15 N14 N15 7    R1    inf  inf   0
N14-N16 N14 N16 4    R1    inf  inf   0
N14-N17 N14 N17 5    R1    inf  inf   0
N14-N18 N14 N18 7    R1    inf  inf   0
N14-N19 N14 N19 9    R1    inf  inf   0
N14-N20 N14 N20 3    R1    inf  inf   0
N15-N16 N15 N16 2    R1    inf  inf   0
N15-N17 N15 N17 2    R1    inf  inf   0
N15-N18 N15 N18 7    R1    inf  inf   0
N15-N19 N15 N19 5    R1    inf  inf   0
N15-N20 N15 N20 9    R1    inf  inf   0
N16-N17 N16 N17 7    R1    inf  inf   0
N16-N18 N16 N18 2    R1    inf  inf   0
N16-N19 N16 N19 6    R1    inf  inf   0
N16-N20 N16 N20 5    R1    inf  inf   0
N17-N18 N17 N18 8    R1    inf  inf   0
N17-N19 N17 N19 2    R1    inf  inf   0
N17-N20 N17 N20 8    R1    inf  inf   0
N18-N19 N18 N19 10    R1    inf  inf   0
N18-N20 N18 N20 9    R1    inf  inf   0
N19-N20 N19 N20 9    R1    inf  inf   0
```

6. 25n50s Network

Topology

NODE	X	Y	SIZE
N01	92	136	3
N02	175	78	3
N03	266	117	3
N04	359	32	3
N05	390	159	3
N06	344	239	5
N07	480	223	4
N08	561	195	4
N09	515	297	3
N10	432	290	6
N11	564	411	5
N12	446	414	4
N13	504	482	3
N14	390	454	4
N15	351	316	7
N16	337	556	3

N17	168	571	4
N18	212	427	5
N19	127	451	4
N20	193	375	3
N21	155	283	6
N22	52	349	4
N23	105	254	3
N24	245	286	4
N25	210	190	4

SPAN	O	D	LENGTH	MTBF	MTTR	FIXEDCOST	UNITCOST
S01	N01	N02	101.257	0	0	0	101.257
S02	N01	N21	159.931	0	0	0	159.931
S03	N02	N03	99.005	0	0	0	99.005
S04	N02	N25	117.341	0	0	0	117.341
S05	N03	N04	125.992	0	0	0	125.992
S06	N03	N06	144.803	0	0	0	144.803
S07	N05	N04	130.729	0	0	0	130.729
S08	N05	N07	110.436	0	0	0	110.436
S09	N06	N24	109.59	0	0	0	109.59
S10	N07	N06	136.938	0	0	0	136.938
S11	N07	N08	85.703	0	0	0	85.703
S12	N07	N10	82.42	0	0	0	82.42
S13	N08	N04	259.563	0	0	0	259.563
S14	N08	N09	111.893	0	0	0	111.893
S15	N08	N10	160.206	0	0	0	160.206
S16	N09	N10	83.295	0	0	0	83.295
S17	N09	N11	124.085	0	0	0	124.085
S18	N10	N06	101.71	0	0	0	101.71
S19	N11	N10	179.067	0	0	0	179.067
S20	N11	N13	92.957	0	0	0	92.957
S21	N11	N15	233.225	0	0	0	233.225
S22	N12	N11	118.038	0	0	0	118.038
S23	N12	N13	89.376	0	0	0	89.376
S24	N13	N14	117.388	0	0	0	117.388
S25	N14	N12	68.819	0	0	0	68.819
S26	N15	N10	85.071	0	0	0	85.071
S27	N15	N12	136.488	0	0	0	136.488
S28	N15	N14	143.405	0	0	0	143.405
S29	N15	N17	313.869	0	0	0	313.869
S30	N16	N14	114.948	0	0	0	114.948
S31	N16	N18	179.627	0	0	0	179.627
S32	N17	N16	169.664	0	0	0	169.664
S33	N17	N19	126.811	0	0	0	126.811
S34	N18	N15	177.882	0	0	0	177.882
S35	N18	N17	150.572	0	0	0	150.572
S36	N19	N18	88.323	0	0	0	88.323
S37	N20	N18	55.362	0	0	0	55.362
S38	N21	N19	170.317	0	0	0	170.317
S39	N21	N22	122.332	0	0	0	122.332
S40	N21	N24	90.05	0	0	0	90.05
S41	N22	N19	126.606	0	0	0	126.606
S42	N22	N20	143.377	0	0	0	143.377
S43	N23	N01	118.714	0	0	0	118.714
S44	N23	N21	57.801	0	0	0	57.801
S45	N23	N22	108.784	0	0	0	108.784
S46	N24	N15	110.164	0	0	0	110.164
S47	N24	N20	103.078	0	0	0	103.078
S48	N25	N05	182.65	0	0	0	182.65
S49	N25	N06	142.678	0	0	0	142.678
S50	N25	N21	108.046	0	0	0	108.046

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
D1	N01	N02	20	R1	inf	inf	0
D2	N01	N03	20	R1	inf	inf	0
D3	N01	N04	20	R1	inf	inf	0

D4	N01	N05	20	R1	inf	inf	0
D5	N01	N06	20	R1	inf	inf	0
D6	N01	N07	20	R1	inf	inf	0
D7	N01	N08	20	R1	inf	inf	0
D8	N01	N09	20	R1	inf	inf	0
D9	N01	N10	20	R1	inf	inf	0
D10	N01	N11	20	R1	inf	inf	0
D11	N01	N12	20	R1	inf	inf	0
D12	N01	N13	20	R1	inf	inf	0
D13	N01	N14	20	R1	inf	inf	0
D14	N01	N15	20	R1	inf	inf	0
D15	N01	N16	20	R1	inf	inf	0
D16	N01	N17	20	R1	inf	inf	0
D17	N01	N18	20	R1	inf	inf	0
D18	N01	N19	20	R1	inf	inf	0
D19	N01	N20	20	R1	inf	inf	0
D20	N01	N21	20	R1	inf	inf	0
D21	N01	N22	20	R1	inf	inf	0
D22	N01	N23	20	R1	inf	inf	0
D23	N01	N24	20	R1	inf	inf	0
D24	N01	N25	20	R1	inf	inf	0
D25	N02	N03	20	R1	inf	inf	0
D26	N02	N04	20	R1	inf	inf	0
D27	N02	N05	20	R1	inf	inf	0
D28	N02	N06	20	R1	inf	inf	0
D29	N02	N07	20	R1	inf	inf	0
D30	N02	N08	20	R1	inf	inf	0
D31	N02	N09	20	R1	inf	inf	0
D32	N02	N10	20	R1	inf	inf	0
D33	N02	N11	20	R1	inf	inf	0
D34	N02	N12	20	R1	inf	inf	0
D35	N02	N13	20	R1	inf	inf	0
D36	N02	N14	20	R1	inf	inf	0
D37	N02	N15	20	R1	inf	inf	0
D38	N02	N16	20	R1	inf	inf	0
D39	N02	N17	20	R1	inf	inf	0
D40	N02	N18	20	R1	inf	inf	0
D41	N02	N19	20	R1	inf	inf	0
D42	N02	N20	20	R1	inf	inf	0
D43	N02	N21	20	R1	inf	inf	0
D44	N02	N22	20	R1	inf	inf	0
D45	N02	N23	20	R1	inf	inf	0
D46	N02	N24	20	R1	inf	inf	0
D47	N02	N25	20	R1	inf	inf	0
D48	N03	N04	20	R1	inf	inf	0
D49	N03	N05	20	R1	inf	inf	0
D50	N03	N06	20	R1	inf	inf	0
D51	N03	N07	20	R1	inf	inf	0
D52	N03	N08	20	R1	inf	inf	0
D53	N03	N09	20	R1	inf	inf	0
D54	N03	N10	20	R1	inf	inf	0
D55	N03	N11	20	R1	inf	inf	0
D56	N03	N12	20	R1	inf	inf	0
D57	N03	N13	20	R1	inf	inf	0
D58	N03	N14	20	R1	inf	inf	0
D59	N03	N15	20	R1	inf	inf	0
D60	N03	N16	20	R1	inf	inf	0
D61	N03	N17	20	R1	inf	inf	0
D62	N03	N18	20	R1	inf	inf	0
D63	N03	N19	20	R1	inf	inf	0
D64	N03	N20	20	R1	inf	inf	0
D65	N03	N21	20	R1	inf	inf	0
D66	N03	N22	20	R1	inf	inf	0
D67	N03	N23	20	R1	inf	inf	0
D68	N03	N24	20	R1	inf	inf	0
D69	N03	N25	20	R1	inf	inf	0
D70	N04	N05	20	R1	inf	inf	0
D71	N04	N06	20	R1	inf	inf	0

D72	N04	N07	20	R1	inf	inf	0
D73	N04	N08	20	R1	inf	inf	0
D74	N04	N09	20	R1	inf	inf	0
D75	N04	N10	20	R1	inf	inf	0
D76	N04	N11	20	R1	inf	inf	0
D77	N04	N12	20	R1	inf	inf	0
D78	N04	N13	20	R1	inf	inf	0
D79	N04	N14	20	R1	inf	inf	0
D80	N04	N15	20	R1	inf	inf	0
D81	N04	N16	20	R1	inf	inf	0
D82	N04	N17	20	R1	inf	inf	0
D83	N04	N18	20	R1	inf	inf	0
D84	N04	N19	20	R1	inf	inf	0
D85	N04	N20	20	R1	inf	inf	0
D86	N04	N21	20	R1	inf	inf	0
D87	N04	N22	20	R1	inf	inf	0
D88	N04	N23	20	R1	inf	inf	0
D89	N04	N24	20	R1	inf	inf	0
D90	N04	N25	20	R1	inf	inf	0
D91	N05	N06	20	R1	inf	inf	0
D92	N05	N07	20	R1	inf	inf	0
D93	N05	N08	20	R1	inf	inf	0
D94	N05	N09	20	R1	inf	inf	0
D95	N05	N10	20	R1	inf	inf	0
D96	N05	N11	20	R1	inf	inf	0
D97	N05	N12	20	R1	inf	inf	0
D98	N05	N13	20	R1	inf	inf	0
D99	N05	N14	20	R1	inf	inf	0
D100	N05	N15	20	R1	inf	inf	0
D101	N05	N16	20	R1	inf	inf	0
D102	N05	N17	20	R1	inf	inf	0
D103	N05	N18	20	R1	inf	inf	0
D104	N05	N19	20	R1	inf	inf	0
D105	N05	N20	20	R1	inf	inf	0
D106	N05	N21	20	R1	inf	inf	0
D107	N05	N22	20	R1	inf	inf	0
D108	N05	N23	20	R1	inf	inf	0
D109	N05	N24	20	R1	inf	inf	0
D110	N05	N25	20	R1	inf	inf	0
D111	N06	N07	20	R1	inf	inf	0
D112	N06	N08	20	R1	inf	inf	0
D113	N06	N09	20	R1	inf	inf	0
D114	N06	N10	20	R1	inf	inf	0
D115	N06	N11	20	R1	inf	inf	0
D116	N06	N12	20	R1	inf	inf	0
D117	N06	N13	20	R1	inf	inf	0
D118	N06	N14	20	R1	inf	inf	0
D119	N06	N15	20	R1	inf	inf	0
D120	N06	N16	20	R1	inf	inf	0
D121	N06	N17	20	R1	inf	inf	0
D122	N06	N18	20	R1	inf	inf	0
D123	N06	N19	20	R1	inf	inf	0
D124	N06	N20	20	R1	inf	inf	0
D125	N06	N21	20	R1	inf	inf	0
D126	N06	N22	20	R1	inf	inf	0
D127	N06	N23	20	R1	inf	inf	0
D128	N06	N24	20	R1	inf	inf	0
D129	N06	N25	20	R1	inf	inf	0
D130	N07	N08	20	R1	inf	inf	0
D131	N07	N09	20	R1	inf	inf	0
D132	N07	N10	20	R1	inf	inf	0
D133	N07	N11	20	R1	inf	inf	0
D134	N07	N12	20	R1	inf	inf	0
D135	N07	N13	20	R1	inf	inf	0
D136	N07	N14	20	R1	inf	inf	0
D137	N07	N15	20	R1	inf	inf	0
D138	N07	N16	20	R1	inf	inf	0
D139	N07	N17	20	R1	inf	inf	0

D140	N07	N18	20	R1	inf	inf	0
D141	N07	N19	20	R1	inf	inf	0
D142	N07	N20	20	R1	inf	inf	0
D143	N07	N21	20	R1	inf	inf	0
D144	N07	N22	20	R1	inf	inf	0
D145	N07	N23	20	R1	inf	inf	0
D146	N07	N24	20	R1	inf	inf	0
D147	N07	N25	20	R1	inf	inf	0
D148	N08	N09	20	R1	inf	inf	0
D149	N08	N10	20	R1	inf	inf	0
D150	N08	N11	20	R1	inf	inf	0
D151	N08	N12	20	R1	inf	inf	0
D152	N08	N13	20	R1	inf	inf	0
D153	N08	N14	20	R1	inf	inf	0
D154	N08	N15	20	R1	inf	inf	0
D155	N08	N16	20	R1	inf	inf	0
D156	N08	N17	20	R1	inf	inf	0
D157	N08	N18	20	R1	inf	inf	0
D158	N08	N19	20	R1	inf	inf	0
D159	N08	N20	20	R1	inf	inf	0
D160	N08	N21	20	R1	inf	inf	0
D161	N08	N22	20	R1	inf	inf	0
D162	N08	N23	20	R1	inf	inf	0
D163	N08	N24	20	R1	inf	inf	0
D164	N08	N25	20	R1	inf	inf	0
D165	N09	N10	20	R1	inf	inf	0
D166	N09	N11	20	R1	inf	inf	0
D167	N09	N12	20	R1	inf	inf	0
D168	N09	N13	20	R1	inf	inf	0
D169	N09	N14	20	R1	inf	inf	0
D170	N09	N15	20	R1	inf	inf	0
D171	N09	N16	20	R1	inf	inf	0
D172	N09	N17	20	R1	inf	inf	0
D173	N09	N18	20	R1	inf	inf	0
D174	N09	N19	20	R1	inf	inf	0
D175	N09	N20	20	R1	inf	inf	0
D176	N09	N21	20	R1	inf	inf	0
D177	N09	N22	20	R1	inf	inf	0
D178	N09	N23	20	R1	inf	inf	0
D179	N09	N24	20	R1	inf	inf	0
D180	N09	N25	20	R1	inf	inf	0
D181	N10	N11	20	R1	inf	inf	0
D182	N10	N12	20	R1	inf	inf	0
D183	N10	N13	20	R1	inf	inf	0
D184	N10	N14	20	R1	inf	inf	0
D185	N10	N15	20	R1	inf	inf	0
D186	N10	N16	20	R1	inf	inf	0
D187	N10	N17	20	R1	inf	inf	0
D188	N10	N18	20	R1	inf	inf	0
D189	N10	N19	20	R1	inf	inf	0
D190	N10	N20	20	R1	inf	inf	0
D191	N10	N21	20	R1	inf	inf	0
D192	N10	N22	20	R1	inf	inf	0
D193	N10	N23	20	R1	inf	inf	0
D194	N10	N24	20	R1	inf	inf	0
D195	N10	N25	20	R1	inf	inf	0
D196	N11	N12	20	R1	inf	inf	0
D197	N11	N13	20	R1	inf	inf	0
D198	N11	N14	20	R1	inf	inf	0
D199	N11	N15	20	R1	inf	inf	0
D200	N11	N16	20	R1	inf	inf	0
D201	N11	N17	20	R1	inf	inf	0
D202	N11	N18	20	R1	inf	inf	0
D203	N11	N19	20	R1	inf	inf	0
D204	N11	N20	20	R1	inf	inf	0
D205	N11	N21	20	R1	inf	inf	0
D206	N11	N22	20	R1	inf	inf	0
D207	N11	N23	20	R1	inf	inf	0

D208	N11	N24	20	R1	inf	inf	0
D209	N11	N25	20	R1	inf	inf	0
D210	N12	N13	20	R1	inf	inf	0
D211	N12	N14	20	R1	inf	inf	0
D212	N12	N15	20	R1	inf	inf	0
D213	N12	N16	20	R1	inf	inf	0
D214	N12	N17	20	R1	inf	inf	0
D215	N12	N18	20	R1	inf	inf	0
D216	N12	N19	20	R1	inf	inf	0
D217	N12	N20	20	R1	inf	inf	0
D218	N12	N21	20	R1	inf	inf	0
D219	N12	N22	20	R1	inf	inf	0
D220	N12	N23	20	R1	inf	inf	0
D221	N12	N24	20	R1	inf	inf	0
D222	N12	N25	20	R1	inf	inf	0
D223	N13	N14	20	R1	inf	inf	0
D224	N13	N15	20	R1	inf	inf	0
D225	N13	N16	20	R1	inf	inf	0
D226	N13	N17	20	R1	inf	inf	0
D227	N13	N18	20	R1	inf	inf	0
D228	N13	N19	20	R1	inf	inf	0
D229	N13	N20	20	R1	inf	inf	0
D230	N13	N21	20	R1	inf	inf	0
D231	N13	N22	20	R1	inf	inf	0
D232	N13	N23	20	R1	inf	inf	0
D233	N13	N24	20	R1	inf	inf	0
D234	N13	N25	20	R1	inf	inf	0
D235	N14	N15	20	R1	inf	inf	0
D236	N14	N16	20	R1	inf	inf	0
D237	N14	N17	20	R1	inf	inf	0
D238	N14	N18	20	R1	inf	inf	0
D239	N14	N19	20	R1	inf	inf	0
D240	N14	N20	20	R1	inf	inf	0
D241	N14	N21	20	R1	inf	inf	0
D242	N14	N22	20	R1	inf	inf	0
D243	N14	N23	20	R1	inf	inf	0
D244	N14	N24	20	R1	inf	inf	0
D245	N14	N25	20	R1	inf	inf	0
D246	N15	N16	20	R1	inf	inf	0
D247	N15	N17	20	R1	inf	inf	0
D248	N15	N18	20	R1	inf	inf	0
D249	N15	N19	20	R1	inf	inf	0
D250	N15	N20	20	R1	inf	inf	0
D251	N15	N21	20	R1	inf	inf	0
D252	N15	N22	20	R1	inf	inf	0
D253	N15	N23	20	R1	inf	inf	0
D254	N15	N24	20	R1	inf	inf	0
D255	N15	N25	20	R1	inf	inf	0
D256	N16	N17	20	R1	inf	inf	0
D257	N16	N18	20	R1	inf	inf	0
D258	N16	N19	20	R1	inf	inf	0
D259	N16	N20	20	R1	inf	inf	0
D260	N16	N21	20	R1	inf	inf	0
D261	N16	N22	20	R1	inf	inf	0
D262	N16	N23	20	R1	inf	inf	0
D263	N16	N24	20	R1	inf	inf	0
D264	N16	N25	20	R1	inf	inf	0
D265	N17	N18	20	R1	inf	inf	0
D266	N17	N19	20	R1	inf	inf	0
D267	N17	N20	20	R1	inf	inf	0
D268	N17	N21	20	R1	inf	inf	0
D269	N17	N22	20	R1	inf	inf	0
D270	N17	N23	20	R1	inf	inf	0
D271	N17	N24	20	R1	inf	inf	0
D272	N17	N25	20	R1	inf	inf	0
D273	N18	N19	20	R1	inf	inf	0
D274	N18	N20	20	R1	inf	inf	0
D275	N18	N21	20	R1	inf	inf	0

D276	N18	N22	20	R1	inf	inf	0
D277	N18	N23	20	R1	inf	inf	0
D278	N18	N24	20	R1	inf	inf	0
D279	N18	N25	20	R1	inf	inf	0
D280	N19	N20	20	R1	inf	inf	0
D281	N19	N21	20	R1	inf	inf	0
D282	N19	N22	20	R1	inf	inf	0
D283	N19	N23	20	R1	inf	inf	0
D284	N19	N24	20	R1	inf	inf	0
D285	N19	N25	20	R1	inf	inf	0
D286	N20	N21	20	R1	inf	inf	0
D287	N20	N22	20	R1	inf	inf	0
D288	N20	N23	20	R1	inf	inf	0
D289	N20	N24	20	R1	inf	inf	0
D290	N20	N25	20	R1	inf	inf	0
D291	N21	N22	20	R1	inf	inf	0
D292	N21	N23	20	R1	inf	inf	0
D293	N21	N24	20	R1	inf	inf	0
D294	N21	N25	20	R1	inf	inf	0
D295	N22	N23	20	R1	inf	inf	0
D296	N22	N24	20	R1	inf	inf	0
D297	N22	N25	20	R1	inf	inf	0
D298	N23	N24	20	R1	inf	inf	0
D299	N23	N25	20	R1	inf	inf	0
D300	N24	N25	20	R1	inf	inf	0

7. 9n16s Network

Topology

NODE	X	Y	SIZE
Node1	216	86	3
Node2	189	132	4
Node3	177	193	4
Node4	160	260	3
Node5	261	358	3
Node6	465	294	3
Node7	598	217	3
Node8	582	94	5
Node9	417	77	4

SPAN	O	D	LENGTH	MTBF	MTTR	FIXEDCOST	UNITCOST
Node1-Node5	Node1	Node5	275.697	0	0	0	275.697
Node2-Node8	Node2	Node8	394.833	0	0	0	394.833
Node2-Node9	Node2	Node9	234.54	0	0	0	234.54
Node3-Node7	Node3	Node7	421.684	0	0	0	421.684
Node3-Node8	Node3	Node8	416.924	0	0	0	416.924
Node4-Node9	Node4	Node9	315.496	0	0	0	315.496
Node4-Node5	Node4	Node5	140.73	0	0	0	140.73
Node4-Node3	Node4	Node3	69.123	0	0	0	69.123
Node6-Node7	Node6	Node7	153.681	0	0	0	153.681
Node8-Node6	Node8	Node6	231.709	0	0	0	231.709
Node5-Node6	Node5	Node6	213.804	0	0	0	213.804
Node8-Node7	Node8	Node7	124.036	0	0	0	124.036
Node9-Node8	Node9	Node8	165.873	0	0	0	165.873
Node9-Node1	Node9	Node1	201.201	0	0	0	201.201
Node2-Node3	Node2	Node3	62.169	0	0	0	62.169
Node2-Node1	Node2	Node1	53.339	0	0	0	53.339

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
Node1-Node2	Node1	Node2	188	R1	inf	inf	0
Node1-Node3	Node1	Node3	100	R1	inf	inf	0
Node1-Node4	Node1	Node4	149	R1	inf	inf	0
Node1-Node5	Node1	Node5	191	R1	inf	inf	0
Node1-Node6	Node1	Node6	115	R1	inf	inf	0

Node1-Node7	Node1	Node7	159	R1	inf	inf	0
Node1-Node8	Node1	Node8	111	R1	inf	inf	0
Node1-Node9	Node1	Node9	140	R1	inf	inf	0
Node2-Node3	Node2	Node3	129	R1	inf	inf	0
Node2-Node4	Node2	Node4	184	R1	inf	inf	0
Node2-Node5	Node2	Node5	150	R1	inf	inf	0
Node2-Node6	Node2	Node6	108	R1	inf	inf	0
Node2-Node7	Node2	Node7	107	R1	inf	inf	0
Node2-Node8	Node2	Node8	177	R1	inf	inf	0
Node2-Node9	Node2	Node9	157	R1	inf	inf	0
Node3-Node4	Node3	Node4	123	R1	inf	inf	0
Node3-Node5	Node3	Node5	196	R1	inf	inf	0
Node3-Node6	Node3	Node6	149	R1	inf	inf	0
Node3-Node7	Node3	Node7	154	R1	inf	inf	0
Node3-Node8	Node3	Node8	160	R1	inf	inf	0
Node3-Node9	Node3	Node9	188	R1	inf	inf	0
Node4-Node5	Node4	Node5	108	R1	inf	inf	0
Node4-Node6	Node4	Node6	195	R1	inf	inf	0
Node4-Node7	Node4	Node7	146	R1	inf	inf	0
Node4-Node8	Node4	Node8	156	R1	inf	inf	0
Node4-Node9	Node4	Node9	195	R1	inf	inf	0
Node5-Node6	Node5	Node6	198	R1	inf	inf	0
Node5-Node7	Node5	Node7	177	R1	inf	inf	0
Node5-Node8	Node5	Node8	158	R1	inf	inf	0
Node5-Node9	Node5	Node9	162	R1	inf	inf	0
Node6-Node7	Node6	Node7	134	R1	inf	inf	0
Node6-Node8	Node6	Node8	174	R1	inf	inf	0
Node6-Node9	Node6	Node9	146	R1	inf	inf	0
Node7-Node8	Node7	Node8	108	R1	inf	inf	0
Node7-Node9	Node7	Node9	114	R1	inf	inf	0
Node8-Node9	Node8	Node9	126	R1	inf	inf	0

8. 7n11s Network

Topology

NODE	X	Y	SIZE
Node1	265	164	3
Node2	193	237	3
Node3	254	341	3
Node4	366	227	3
Node5	658	185	3
Node6	593	369	4
Node7	500	74	3

SPAN	O	D	LENGTH	MTBF	MTTR	FIXEDCOST	UNITCOST
Node1-Node4	Node1	Node4	119.038	0	0	0	119.038
Node7-Node6	Node7	Node6	309.312	0	0	0	309.312
Node7-Node5	Node7	Node5	193.093	0	0	0	193.093
Node5-Node6	Node5	Node6	195.144	0	0	0	195.144
Node4-Node5	Node4	Node5	295.005	0	0	0	295.005
Node1-Node7	Node1	Node7	251.645	0	0	0	251.645
Node3-Node4	Node3	Node4	159.812	0	0	0	159.812
Node2-Node3	Node2	Node3	120.569	0	0	0	120.569
Node2-Node1	Node2	Node1	102.533	0	0	0	102.533
Node3-Node6	Node3	Node6	340.154	0	0	0	340.154
Node2-Node6	Node2	Node6	421.217	0	0	0	421.217

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
Node1-Node2	Node1	Node2	124	R1	inf	inf	0
Node1-Node3	Node1	Node3	107	R1	inf	inf	0
Node1-Node4	Node1	Node4	182	R1	inf	inf	0
Node1-Node5	Node1	Node5	125	R1	inf	inf	0
Node1-Node6	Node1	Node6	173	R1	inf	inf	0
Node1-Node7	Node1	Node7	100	R1	inf	inf	0

Node2-Node3	Node2	Node3	151	R1	inf	inf	0
Node2-Node4	Node2	Node4	168	R1	inf	inf	0
Node2-Node5	Node2	Node5	187	R1	inf	inf	0
Node2-Node6	Node2	Node6	124	R1	inf	inf	0
Node2-Node7	Node2	Node7	128	R1	inf	inf	0
Node3-Node4	Node3	Node4	100	R1	inf	inf	0
Node3-Node5	Node3	Node5	108	R1	inf	inf	0
Node3-Node6	Node3	Node6	149	R1	inf	inf	0
Node3-Node7	Node3	Node7	140	R1	inf	inf	0
Node4-Node5	Node4	Node5	123	R1	inf	inf	0
Node4-Node6	Node4	Node6	148	R1	inf	inf	0
Node4-Node7	Node4	Node7	144	R1	inf	inf	0
Node5-Node6	Node5	Node6	124	R1	inf	inf	0
Node5-Node7	Node5	Node7	110	R1	inf	inf	0
Node6-Node7	Node6	Node7	103	R1	inf	inf	0

9. TELUS Metro Calgary Network (Isomorphic)

Topology

NODE	X	Y	SIZE
A	519	381	0
B	667	901	0
C	60	458	0
D	686	75	0
E	260	394	0
F	520	69	0
G	58	918	0
H	135	49	0
I	751	148	0
K	568	206	0
L	332	753	0
M	456	752	0
N	25	640	0
O	117	785	0
P	477	1070	0
Q	549	623	0
R	717	1007	0
S	70	121	0
T	149	364	0
U	428	1086	0
V	226	996	0
W	373	87	0
X	829	561	0
Y	243	848	0
Z	768	263	0
AA	309	163	0
BB	308	142	0
CC	244	595	0
DD	367	1076	0
EE	540	802	0
FF	687	354	0
GG	803	218	0
HH	219	890	0
AE	412	32	2
CE	541	30	2
KK	266	160	2
II	750	380	2

SPAN	O	D	LENGTH	MTBF	MTTR	FIXEDCOST	UNITCOST
S1	A	G	7.3	0	0	0	7.3
S2	A	K	3.7	0	0	0	3.7
S3	A	T	3.2	0	0	0	3.2
S4	A	X	4	0	0	0	4
S5	A	AA	3.1	0	0	0	3.1
S6	B	L	11.2	0	0	0	11.2
S7	B	P	5.6	0	0	0	5.6

S8	B	Q	3.7	0	0	0	3.7
S9	B	X	6.8	0	0	0	6.8
S10	B	EE	6.5	0	0	0	6.5
S11	C	N	5.5	0	0	0	5.5
S12	C	S	13.8	0	0	0	13.8
S13	C	T	7	0	0	0	7
S14	D	F	7	0	0	0	7
S15	D	AA	8.3	0	0	0	8.3
S16	E	H	9.4	0	0	0	9.4
S17	E	T	3	0	0	0	3
S18	E	AA	2.5	0	0	0	2.5
S20	F	W	7.4	0	0	0	7.4
S22	G	N	12.2	0	0	0	12.2
S23	G	O	6.4	0	0	0	6.4
S24	G	CC	5.1	0	0	0	5.1
S25	H	S	5.4	0	0	0	5.4
S26	H	T	6.5	0	0	0	6.5
S27	H	W	6.1	0	0	0	6.1
S28	I	K	2.8	0	0	0	2.8
S30	I	FF	6.8	0	0	0	6.8
S31	K	X	3.8	0	0	0	3.8
S32	K	AA	2	0	0	0	2
S33	L	P	17	0	0	0	17
S34	L	V	5.2	0	0	0	5.2
S35	L	Y	2.6	0	0	0	2.6
S36	L	HH	4.9	0	0	0	4.9
S37	M	P	6.8	0	0	0	6.8
S38	M	EE	11.8	0	0	0	11.8
S39	Q	X	4.4	0	0	0	4.4
S40	R	EE	5.9	0	0	0	5.9
S41	T	CC	4.8	0	0	0	4.8
S42	X	Z	4.1	0	0	0	4.1
BB-F	BB	F	195.451	0	0	0	195.451
BB-H	BB	H	85.615	0	0	0	85.615
Y-G	Y	G	150.659	0	0	0	150.659
I-GG	I	GG	152.761	0	0	0	152.761
GG-X	GG	X	169.452	0	0	0	169.452

Demands

DEMAND	O	D	NBUNITS	RESTCLASS	HOPLIM	DISTLIM	MAXOUTAGE
D1	A	B	62	R1	inf	inf	0
D2	A	C	42	R1	inf	inf	0
D3	A	D	15	R1	inf	inf	0
D4	A	E	6	R1	inf	inf	0
D5	A	F	21	R1	inf	inf	0
D6	A	G	26	R1	inf	inf	0
D7	A	H	40	R1	inf	inf	0
D8	A	I	25	R1	inf	inf	0
D9	A	K	20	R1	inf	inf	0
D10	A	L	33	R1	inf	inf	0
D11	A	M	16	R1	inf	inf	0
D12	A	N	13	R1	inf	inf	0
D13	A	P	4	R1	inf	inf	0
D14	A	Q	4	R1	inf	inf	0
D15	A	T	4	R1	inf	inf	0
D16	A	X	3	R1	inf	inf	0
D17	A	AA	12	R1	inf	inf	0
D18	A	CC	13	R1	inf	inf	0
D19	A	EE	1	R1	inf	inf	0
D20	A	KK	14	R1	inf	inf	0
D21	A	II	10	R1	inf	inf	0
D22	A	Z	2	R1	inf	inf	0
D23	B	C	1	R1	inf	inf	0
D24	B	D	23	R1	inf	inf	0
D25	B	E	5	R1	inf	inf	0
D26	B	F	1	R1	inf	inf	0
D27	B	G	3	R1	inf	inf	0

D28	B	H	5	R1	inf	inf	0
D29	B	I	2	R1	inf	inf	0
D30	B	K	15	R1	inf	inf	0
D31	B	L	16	R1	inf	inf	0
D32	B	M	1	R1	inf	inf	0
D33	B	N	1	R1	inf	inf	0
D34	B	P	1	R1	inf	inf	0
D35	B	Q	11	R1	inf	inf	0
D36	B	T	1	R1	inf	inf	0
D37	B	X	1	R1	inf	inf	0
D38	B	AA	2	R1	inf	inf	0
D39	B	DD	1	R1	inf	inf	0
D40	B	EE	11	R1	inf	inf	0
D41	B	Z	2	R1	inf	inf	0
D42	C	G	1	R1	inf	inf	0
D43	C	H	6	R1	inf	inf	0
D44	C	K	2	R1	inf	inf	0
D45	C	N	10	R1	inf	inf	0
D46	C	T	5	R1	inf	inf	0
D47	C	AA	1	R1	inf	inf	0
D48	D	F	7	R1	inf	inf	0
D49	D	H	1	R1	inf	inf	0
D50	D	K	5	R1	inf	inf	0
D51	D	T	1	R1	inf	inf	0
D52	D	AA	1	R1	inf	inf	0
D53	D	BB	6	R1	inf	inf	0
D54	D	II	1	R1	inf	inf	0
D55	E	H	2	R1	inf	inf	0
D56	E	I	1	R1	inf	inf	0
D57	E	N	1	R1	inf	inf	0
D58	E	X	1	R1	inf	inf	0
D59	E	AA	1	R1	inf	inf	0
D60	F	H	4	R1	inf	inf	0
D61	F	T	1	R1	inf	inf	0
D62	F	AA	1	R1	inf	inf	0
D63	G	L	10	R1	inf	inf	0
D64	G	N	5	R1	inf	inf	0
D65	G	P	1	R1	inf	inf	0
D66	G	X	1	R1	inf	inf	0
D67	G	CC	7	R1	inf	inf	0
D68	H	T	1	R1	inf	inf	0
D69	H	X	6	R1	inf	inf	0
D70	H	AA	1	R1	inf	inf	0
D71	H	BB	6	R1	inf	inf	0
D72	H	KK	12	R1	inf	inf	0
D73	I	K	6	R1	inf	inf	0
D74	I	M	1	R1	inf	inf	0
D75	I	X	1	R1	inf	inf	0
D76	I	AA	1	R1	inf	inf	0
D77	I	GG	4	R1	inf	inf	0
D78	K	T	2	R1	inf	inf	0
D79	K	X	3	R1	inf	inf	0
D80	K	AA	5	R1	inf	inf	0
D81	K	II	7	R1	inf	inf	0
D82	L	M	3	R1	inf	inf	0
D83	L	N	6	R1	inf	inf	0
D84	L	P	9	R1	inf	inf	0
D85	L	V	4	R1	inf	inf	0
D86	L	Y	8	R1	inf	inf	0
D87	L	DD	6	R1	inf	inf	0
D88	L	EE	1	R1	inf	inf	0
D89	M	P	6	R1	inf	inf	0
D90	M	EE	6	R1	inf	inf	0
D91	P	DD	1	R1	inf	inf	0
D92	P	EE	6	R1	inf	inf	0
D93	T	AA	2	R1	inf	inf	0
D94	T	CC	1	R1	inf	inf	0